Optimized Routing of Unmanned Aerial Systems to Address Informational Gaps in Counterinsurgency

by

Andrew C. Lee

B.S. Management
United States Military Academy, 2003

Submitted to the Department of Civil and Environmental Engineering
In Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN TRANSPORTATION
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2012

Signature of Author: ________________________________
Department of Civil and Environmental Engineering
May 11, 2012

Certified by: ________________________________
John M. Irvine
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: ________________________________
Professor Cynthia Barnhart
Professor of Civil and Environmental Engineering and Engineering Systems
Associate Dean for Academic Affairs, School of Engineering
Thesis Supervisor

Accepted by: ________________________________
Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

[This Page Intentionally Left Blank]

Optimized Routing of Unmanned Aerial Systems to Address Informational Gaps in Counterinsurgency

by

Andrew C. Lee

Submitted to the Department of Civil and Environmental Engineering
on May 11, 2012 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Transportation

Recent military conflicts reveal that the ability to assess and improve the health of a society contributes more to a successful counterinsurgency (COIN) than direct military engagement. In COIN, a military commander requires maximum situational awareness not only with regard to the enemy but also to the status of logistical support concerning civil security operations, governance, essential services, economic development, and the host nation's security forces. Although current Brigade level Unmanned Aerial Systems (UAS) can provide critical unadulterated views of progress with respect to these Logistical Lines of Operation (LLO), the majority of units continue to employ UASs for strictly conventional combat support missions. By incorporating these LLO targets into the mission planning cycle with a collective UAS effort, commanders can gain a decisive advantage in COIN. Based on the type of LLO, some of these targets might require more than a single observation to provide the maximum benefit. This thesis explores an integer programming and metaheuristic approach to solve the *Collective UAS Planning Problem* (CUPP). The solution to this problem provides optimal plans for multiple sortie routes for heterogeneous UAS assets that collectively visit these diverse secondary LLO targets while in transition to or from primary mission targets.

By exploiting the modularity of the Raven UAS asset, we observe clear advantages, with respect to the total number of targets observed and the total mission time, from an exchange of Raven UASs and from collective sharing of targets between adjacent units. Comparing with the status quo of decentralized operations, we show that the results of this new concept demonstrate significant improvements in target coverage. Furthermore, the use of metaheuristics with a *Repeated Local Search* algorithm facilitates the fast generation of solutions, each within 1.72% of optimality for problems with up to 5 UASs and 25 nodes. By adopting this new paradigm of collective Raven UAS operations and LLO integration, Brigade level commanders can maximize the use of organic UAS assets to address the complex information requirements characteristic of COIN. Future work for the CUPP to reflect a more realistic model could include the effects of random service times and high priority pop-up targets during mission execution.

Thesis Supervisor: Cynthia Barnhart
Title: Professor of Civil and Environmental Engineering and Engineering Systems
Associate Dean for Academic Affairs, School of Engineering

Technical Supervisor: John M. Irvine
Title: Signal & Image Exploitation Engineer, Charles Stark Draper Laboratory, Inc

ACKNOWLEDGEMENTS

First, I want to praise God for providing me with this unique opportunity while serving in the Army to complete my Graduate studies at MIT. I also want to thank all of my friends, family, and mentors who helped me make the right decisions and at times to take the path less traveled by.

I owe the most to Dr. John Irvine, Dr. Steve Kolitz, and Dr. Cynthia Barnhart. I am indebted to Dr. John Irvine for dedicating his time each week to bestow his expertise, insight, and guidance on the arduous thesis writing process. Your feedback provided a steady guide rail to help direct and focus my work. I cannot thank Dr. Kolitz enough for his help in securing my Draper Laboratory Fellowship, facilitating my acceptance to MIT, and providing feedback on my thesis. I am forever grateful to Dr. Cynthia Barnhart for taking the time from her busy schedule as Dean of Engineering and other responsibilities to thoroughly review my thesis and provide invaluable mentorship.

I want to thank Dr. David Carter and Dr. Mark Abramson from Draper Laboratory for helping me with my Matlab questions and in developing my thesis topic. I also want to thank my long time friend CW3 Jeffrey Stokes for entertaining my initial calls for thesis ideas.

I also want to thank my gifted and cerebral friends at MIT, Vikrant, Ta, Janet, Brian, Matt, Jane, Dave, and Mike, who were always available when I needed help on a problem set or thesis formulation. Mike, we did it buddy; thank you for being there with me each step of the way.

Most importantly, I am so very grateful for my beautiful wife Hana and my precious newborn daughter Emanuela, whose birth on March 18, 2012 provided the impetus to complete my thesis on time. Hana, I love you for your immeasurable understanding, patience, and constant encouragement to strive for greatness in whatever I do.

Finally, I want to thank the Charles Stark Draper Laboratory, MIT, the Omar Bradley Fellowship, the U.S. Army, and the Math Department at the U.S. Military Academy for helping me to secure this memorable assignment in Boston.

Andrew C. Lee, MAJ, USA May 11, 2012

Table of Contents

List of Figures

List of Tables

List of Acronyms

BCT	Brigade Combat Team
BDA	Battle Damage Assessment
BSB	Brigade Support Battalion
BSTB	Brigade Special Troops Battalion
C-IDF	Counter Indirect Fire
C-IED	Counter Improvised Explosive Device
CF	Coalition Force
CM	Collection Management
COIN	Counterinsurgency
CONOP	Concept of Operation
CUPP	Collective UAS Planning Problem
DE	Desired Effects
DoD	Department of Defense
EO	Electro-optical
ETIOV	Earliest Time Information of Value
EWG	Effects Working Group
HN	Host Nation
HVT	High Value Target
IED	Improvised Explosive Device
IMINT	Imagery Intelligence
ILS	Iterative Local Search
IO	Information Operations
IR	Intelligence Requirement
IR	Infrared
ISM	Intelligence, Surveillance, and Reconnaissance Synchronization Matrix
ISR	Intelligence, Surveillance, and Reconnaissance
IW	Irregular Warfare
JOC	Joint Operating Concept
LLO	Lines of Operation
LTIOV	Latest Time Information of Value
MICO	Military Intelligence Company
MiTT	Military Transition Team
NAI	Named Area of Interest
NIIRS	National Imagery Interpretability Rating Scale
ODIN	Observe, Detect, Identify, Neutralize
OP	Orienteering Problem

OPL	Optimization Programming Language
OSRVT	One System Remote Video Terminal
PIR	Priority Intelligence Requirement
PRT	Provincial Reconstruction Team
PSYOP	Psychological Operations
PTT	Police Transition Team
QRF	Quick Reaction Force
RFI	Request for Information
SEAD	Suppression and Destruction of Enemy Air Defense
SIGINT	Signals Intelligence
SIR	Specific Information Requirement
TALS	Tactical Automated Landing System
TIC	Troops in Contact
TOP	Team Orienteering Problem
TOPTW	Team Orienteering Problem with Time Windows
TS	Tabu Search
TST	Time Sensitive Target
TSP	Traveling Salesman Problem
TUAS	Tactical Unmanned Aerial System
UAS	Unmanned Aerial System
VRPTW	Vehicle Routing Problem with Time Windows

1 Introduction

A defining strategy in the wars in Iraq and Afghanistan, counterinsurgency (COIN) is broadly defined as "comprehensive civilian and military efforts taken to simultaneously defeat and contain insurgency and address its root causes" [35]. In short, U.S. forces must ensure that the host nation achieves functional governance and legitimacy to carry out key tasks while enforcing the rule of law. In order to achieve this goal, the U.S. Commander requires maximum situational awareness not only with regard to the enemy but also concerning the status of five Logical Lines of Operation (LLO) that typify logistical support during COIN: 1) Conduct combat operations/civil security operations, 2) Train and employ Host Nation (HN) security forces, 3) establish or restore essential services, 4) support development of better governance, and 5) support economical development [9]. According to the current COIN doctrine, the U.S. Commander must pursue all five simultaneously to stamp out the root cause of insurgency.

Intelligence Requirements (IRs) fill gaps in the command's knowledge and understanding of the battlefield and threat forces. Based on the Commander's decision points and priorities for mission success, he can promote some of these IRs into Priority Intelligence Requirements (PIRs) to properly align intelligence assets [11]. Unlike conventional warfare, questions about the population and the host nation dominate the intelligence requirements in the latter stages of counterinsurgency. Conventional warfare IRs like 'when will the enemy reconnaissance forces cross this border?' get supplanted with questions like 'why is the water sewage plant not operational?' Most importantly, consistent updates from collection assets about the five LLOs can improve the Commander's situational awareness to support key decisions.

Unmanned Aerial Systems (UASs) operate as the eyes of the Commander to "see first, understand first, and act first, decisively" [12]. As the COIN effort hands more responsibility of governance and security to the Host Nation, unit commanders can adjust the focus of UASs to monitor the status of the five LLOs in addition to other primary mission requirements.

With expanding roles across every spectrum in civil and military applications, the Unmanned Aerial System (UAS) provides an efficient and cost effective alternative that averts danger to human life. As UAS technology and its safety record continues to improve, widespread interest grows for its employment in aerial photography, surveillance of land and crops, monitoring of forest fires and environmental conditions, and the recent protection of borders and ports with the Department of Homeland Security. Recent evidence of the embrace of UAS technology includes the 2009 integration of UAS programs into the U.S. Air Force Academy's curriculum, the 2008 creation of the Federal Aviation Administration's Aviation Rulemaking Committee to regulate small UASs, and the increasing reliance of armed UASs for precision strikes across the world.

With the prevalence and expansion of the UAS, doctrinal models as it relates to particular applications will continue to change according to the newest technology available and lessons learned. This thesis explores one possible paradigm shift using a collective approach. Due to the modularity and relative simplicity of the Raven UAS asset, collaborative sharing of targets and bases with adjacent units creates additional opportunities by minimizing travel time for each UAS. We use an integer programming and heuristic approach to plan multiple routes for UAS assets in order to optimally address secondary LLO targets while in transition to or from primary mission targets. Based on the type of LLO, multiple looks at a particular target can enhance mission success rates with redundancy, especially with targets that exhibit high false positive rates.

1.1 Overview

The organization of this thesis follows the development of our UAS route planner from initial concept development to testing and analysis of the final string model that incorporates multiple sorties for each UAS. Following this chapter's summary of the contributions and motivations of our work, Chapters 2 through 7 provide the following:

Chapter 2: Operational Problem. This chapter first provides an overview of Counterinsurgency and the process of Intelligence, Surveillance, and Reconnaissance Synchronization, both from the doctrinal perspective and the current operational framework. We then describe the role of the Effects Working Group, a collection of representatives working on the various facets of COIN's five LLOs and its respective functional areas, from Civil Affairs to Essential Services and Governance. Furthermore, the chapter explores current UAS roles and missions as well as anticipated roles in the latter stages of Counterinsurgency. By tying together the requirements from the Effects Working Group and primary mission requirements, we describe the operational problem and the opportunities available for a collective UAS effort in COIN. We define our problem as the Collective UAS Planning Problem (CUPP).

Chapter 3: Model Development. By providing the assumptions, inputs, outputs, and structure of our problem, this chapter provides the framework for a literature review of similar work in order to properly define our problem. After exploring well researched problems like the Traveling Salesman Problem, the Vehicle Routing Problem, and the Aircraft Routing Problem, we find that the most suitable problem definition for the CUPP combines opportunistic targets using the Team Orienteering Problem with Time Windows (TOPTW) method and mandatory primary targets using traditional routing problem methods. Additionally, due to the high computational costs of relevant combinatorial optimization problems, we explore well known heuristics like Simulated Annealing, Tabu Search, and Iterative Local Search to apply in solving the CUPP.

Chapter 4: Problem Formulation. This chapter presents the integer programming formulations for both the Status Quo concept that reflects current operations and the Swap and Share concept that includes our new paradigm in allowing the swapping of bases and sharing of targets. In order to minimize total mission time and wait times at each node, we use sequential multiple objective optimization. The metaheuristics used to solve both concepts include combinations of the following: insertion, 2 opt, deletion-insertion, 2 exchange, and a base swap heuristic.

Chapter 5: Testing and Analysis. We test five hypotheses in this chapter focusing on the advantages of swapping bases and collective sharing of Raven UAS targets. With implementation of the integer programs in IBM's ILOG OPL and the heuristics for each concept in Matlab, we find that the time required reaching the exact solution with the Mixed Integer Program increases as a function of the number of targets. We find that the Swap and Share concept results in increased objective function values over the Status Quo concept especially for cases involving more than 4 UASs.

Chapter 6: String Extension to Swap and Share Concept. This chapter introduces the string concept for the Swap and Share concept to handle multiple sorties for each UAS as well as the results of a final experiment using pseudorandom data. We find that the number of UASs in the problem should dictate the parameter setting for the maximum number of string sets generated. The largest test case involving 5 UASs and 30 target nodes with 100 string sets generated solved the problem in 1.56 minutes with an objective value within 4% of the maximum optimized value attained.

Chapter 7: Conclusions and Future Work. This chapter summarizes our contributions and findings. We suggest improvements to our metaheuristics and propose the integration of additional types of targets, conditional swapping of bases, immediate pop up targets, and random service times as future work.

1.2 Contributions

The primary contribution of this thesis to the UAS routing and planning literature lies in the domain of preplanning multiple sortie missions, not in the domain of satisfying dynamic requirements while in flight. This research paper contributes the following:

1. A mixed integer programming formulation for the CUPP using both the Status Quo and Swap and Share concepts implemented in IBM ILOG OPL that reveals the gains in target coverage by avoiding stovepipe plans and allowing target and base sharing with respect to Raven UASs.
2. The development of metaheuristics implemented in Matlab for both concepts to solve the CUPP.
3. The development of a string model extension to solve for multiple sorties using metaheuristics implemented in Matlab that will provide maximum

value to the Brigade Commander within a timeframe that meets practical operational constraints.

4. Computational studies and results from applying both concepts along with the string model extension for realistic scenarios requiring tuning of parameters.

1.3 Motivation

Feedback from UAS units returning from tours of duty in Afghanistan and Iraq details misallocation of UAS assets as well as missed opportunities and lengthy idle time periods. From the standpoint of war, missed UAS target opportunities impede the Commander's constant struggle to maintain information supremacy, especially in asymmetric warfare. Additionally, based on the accelerating incorporation of new technologies into UAS assets, we find it important to consolidate the most recent technological gains and to adjust standard operating procedure to maximize results. This thesis focuses on a different paradigm with the Swap and Share Concept, blurring the ownership of target sets and UAS assets in order to maximize the use of UASs, particularly in a counterinsurgency operation. In addition, only a few research papers focus on the maximum use of UASs specifically for Counterinsurgency (COIN) operations, so this area of study provides opportunities for new interpretations.

By drawing on various fields of operations research, this thesis describes a technical approach to enhance a given initial multi-route schedule for a diverse set of UASs by maximizing the observation of a diverse set of COIN targets. This planning tool will account for different characteristics of these targets including time windows, dual visit options for nodes, and shared targets between UASs.

2 Operational Problem

This chapter presents the primary tenets of COIN and lays the groundwork for the development of secondary target sets that present targets of opportunity. Additionally, we show how the Intelligence, Surveillance, and Reconnaissance (ISR) Synchronization process creates Specific Information Requirements (SIRs) for collection assets like the UAS. These SIRs will later take the form of inputs in the concepts presented in chapter 3. We present the current operations framework and opportunities for improvement in COIN applications for the Brigade level Shadow Tactical UAS and Raven UAS systems. The chapter concludes with the research goals for this thesis.

2.1 Counterinsurgency

As the Quadrennial Defense Review Report for 2010 makes clear, the Department of Defense (DoD) continues to rebalance the armed forces in order to retain the capability to conduct large-scale counterinsurgency operations. While counterinsurgency (COIN) can be defined as a struggle for popular support, its counterpart, insurgency, is defined as "the organized use of subversion and violence to seize, nullify, or challenge political control of a region" [35]. Counterinsurgency and insurgency both reside under the category of irregular warfare, a style of war that favors indirect approaches in order to "erode an adversary's power, influence, and will" [18].

2.1.1 Logical Lines of Operation in COIN

According to the Counterinsurgency Field Manual, Commanders use Logistical Lines of Operation (LLOs) in order to conceptualize how to synchronize operations with the Host Nation (HN) to undermine the insurgency while legitimizing the HN government. Five interrelated LLOs that describe current COIN operations are: 1) Conduct combat operations/civil security operations, 2) Train and employ HN security forces, 3) establish or restore essential services, 4) support development of better governance, and 5) support economical development. In addition, Information Operations (IO) supplements the overall COIN effort by emphasizing successes and immediately addressing potential risks. Figure 1 depicts an example of a broad LLO strategy that Commanders at all echelons use to achieve unity of effort [9].

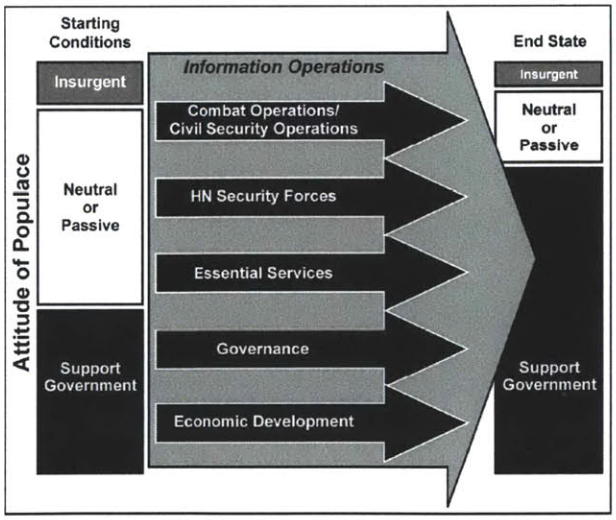


Figure 1 - Example Logistical Lines of Operations. The five interrelated LLOs are shown in the arrows to help bolster local support for the government

2.1.2 Stages of COIN

The overarching goal in COIN "is not to reduce violence to zero or to kill every insurgent, but rather to return the overall system to normality — noting that 'normality' in one society may look different from normality in another" [22]. Commanders should be aware of the stage of COIN they are currently involved with in order to properly align COIN activities. COIN progresses through three indistinct stages analogous to medical lingo: 1) Stop the Bleeding, 2) Inpatient care - Recovery, and 3) Outpatient care - movement to self-sufficiency [9]. During the first stage, it is essential to stop the bleeding by protecting the population and breaking the insurgents' initiative and momentum. During the second stage, the counterinsurgent force must aggressively engage all LLOs to set the HN up for a long-term recovery and restoration of health. Partnership with the host nation is critical. During the last stage, the goal is to transition the responsibility for COIN operations and LLOs to the HN leadership. This thesis focuses on how to effectively employ UASs to engage the LLOs during the second stage of COIN.

2.1.3 Defining Success in COIN

The U.S. Counterinsurgency guide states that a COIN effort could be deemed successful if the counterinsurgent meets the following conditions or Desired Effects (DE):

1. The population views the affected government as legitimate, controlling social, political, economic, and security institutions that meet their needs, including adequate mechanisms to address the grievances that may have fueled support of the insurgency.
2. The insurgent movements and their leaders are co-opted, marginalized, or separated from the population.

3. Armed insurgent forces have dissolved or been demobilized, and/or reintegrated into the political, economic, and social structures of the country [35].

Critical to success, the counterinsurgent must possess detailed knowledge and understanding of the environment, not only concerning the enemy, but also regarding the population. This entails knowing the profile and capabilities of the insurgent groups, the current government, security forces, concerns of the local population, economic status, essential services, sociocultural factors, and infrastructural developments and needs. Understanding the complexity of COIN, the U.S. Army's Counterinsurgency Field Manual FM 3-24 emphasizes the notion that "effective COIN operations are decentralized…higher commanders owe it to their subordinates to push as many capabilities as possible down to their level" [9]. This includes collection assets like Unmanned Aerial Systems (UAS), an Imagery Intelligence (IMINT) platform currently available to the Brigade Combat Team (BCT).

2.2 Unmanned Aerial Systems

From its inception during the Civil War as an unmanned aerial bomber, consisting of a hot air balloon and basket full of timed explosives, Unmanned Aerial Systems (UASs) currently support units in every combat theater. First proving its worth as combat training tools during World War I, UASs served as stealth surveillance assets in Vietnam with the advent of the AQM-34 Ryan Firebee. Today's diverse UASs perform reconnaissance missions with expanded roles in electronic attacks, strike missions, suppression and destruction of enemy air defense (SEAD), network node or communications relay, combat search and rescue, and Signals Intelligence (SIGINT) targeting.

2.2.1 Current UAS Role Shortage and Opportunities

Most of the current UAS missions involve conventional and potentially kinetic combat support roles as opposed to intelligence collection for LLO related targets. Currently UASs in Iraq and Afghanistan associated with Task Force ODIN (standing for "Observe, Detect, Identify, and Neutralize") systematically target Improvised Explosive Devices (IEDs) along key routes used by Coalition Forces. Other missions for UASs in theater include counter-Improvised Explosive Device (C-IED) and counter-Indirect Fire (C-IDF) missions, raids against High Value Targets (HVTs), area surveillance or security, and Troops in Contact (TIC) support missions. As evident from the types of current missions, most of the emphasis for UAS employment centers on finding, fixing, and finishing the enemy while protecting the force. Yet while enroute to these missions, units often disregard the vast amount of imagery potentially vital to the COIN fight. In fact, most units and UAS operators ignore the payload feed while enroute to its primary missions and rarely exploit post mission imagery and video due to staffing and resource limitations. The ability to seamlessly integrate multiple secondary targets for the UAS while enroute to its primary target would provide tremendous added value especially when dealing with targets related to the LLOs, which for the most part require only a cursory look. Additionally, these secondary LLO targets offer a backup collection plan for units who struggle at times with how to utilize the UAS upon early completion of a mission or a canceled mission. An optimized route path will not only expand the use of the UAS, but it will also provide situational awareness of the COIN environment for the U.S. Brigade level Commander, facilitating critical decisions for success.

2.2.2 UAS Indirect Advantages

While ground units with direct access provide invaluable intelligence, the UAS offers a distinct advantage with its remote observation. Military Transition Teams (MiTTs), Police Transition Teams (PTTs), Provincial Reconstruction Teams (PRTs), Civil Affairs Teams, and maneuver combat patrols provide direct feedback from the ground on the progress with the five LLOs. MiTTs address SIRs that in aggregate should answer the IR 'Why are Host Nation security forces not able to take control of operations?' PRTs address IRs like: 'Why is the economy not flourishing in this specific region?' However due to resource limits and security constraints, these teams do not stay with their HN counterparts all day. In monitoring the quality of Host Nation checkpoints and combat patrols, status of essential service infrastructure, and economic activity in marketplaces, the UAS offers an unadulterated view of progress with respect to the five LLOs. Something as simple as trash pickup frequencies, level of pedestrian traffic in markets, and power plant security levels or lack thereof, can trigger pivotal decision points for the Commander based on imagery and Full Motion Video (FMV). Observations of checkpoints or host nation security patrol operations can confirm or deny possible corruption. Additionally, mixing and cueing opportunities with ground assets can provide a more complete understanding of the environment.

2.2.3 Brigade Combat Team UAS Types

A typical Infantry Brigade Combat Team (BCT), a deployable maneuver unit in the US Army, consists of two Battalions of Infantry and a Cavalry, Field Artillery, Special Troops, and Support Battalions. Organic UAS assets within the BCT include the RQ-7B Shadow Tactical Unmanned Aerial System (TUAS) platoon controlled at the Brigade level and the RQ-11B Raven UAS controlled at the Company level. Figure 2 shows the organization and typical distribution of

Shadow and Raven UAS systems in an Infantry Brigade Combat Team. The Shadow TUAS falls under the Military Intelligence Company (MICO) and supports the entire Brigade. Although dependent on the type of unit, each BCT on average receives 15 Raven systems (45 aircraft), which the Commander apportions appropriately to the Battalions. In Figure 2 from left to right, the Brigade Special Troops Battalion (BSTB) owns 1 Raven UAS, the Cavalry Battalion owns 3 Raven UAS, the Infantry Battalion owns 8 Raven UAS, the Fires Battalion owns 2 Raven UAS, and the Brigade Support Battalion (BSB) owns 1 Raven UAS [5]. The Army developed both of the UASs to provide the BCT and Company level elements with responsive tactical intelligence, reconnaissance, and surveillance capability.

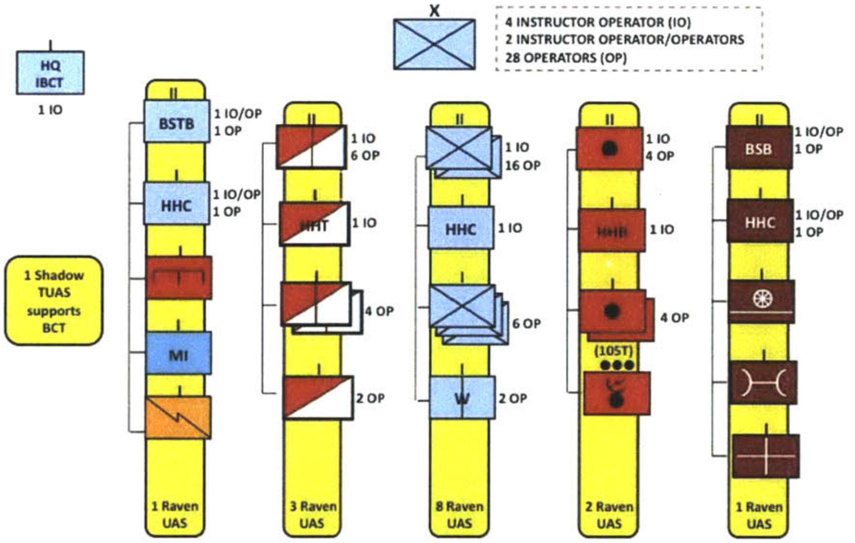


Figure 2 - Brigade Organization Chart. 1 Shadow TUAS system supports the entire Brigade, while anywhere from 1-8 Raven systems support each Battalion (shown in columns) depending on their respective roles (combat or support)

2.2.3.1 Shadow RQ-7B Tactical Unmanned Aerial System

The Shadow RQ-7B TUAS platoon consists of 4 unmanned aircraft, 27 Soldiers and systems including ground control shelters, antenna systems, Tactical Automated Landing Systems (TALS), launcher, and ten Light Tactical

Vehicles. Its capabilities include its dual Electro-optical (EO) and Infrared (IR) sensor payload, ability to laser illuminate with IR to communicate targets to ground maneuver forces, 9 hour endurance, 60 km radius or 125 km range, and airspeed of 60 knots loiter to a maximum of 105 knots (approximately 70 mph to 120 mph). The system also comes with 4 One System Remote Video Terminal (OSRVTs) to provide ground maneuver forces direct feed from the UAS. With a split-based operational capability, mission commanders and pilots can operate the Shadow TUAS system from a forward site and hand off the TUAS to a separate launch and recovery site. The main components of the RQ-7B Shadow TUAS are shown in Figure 3.

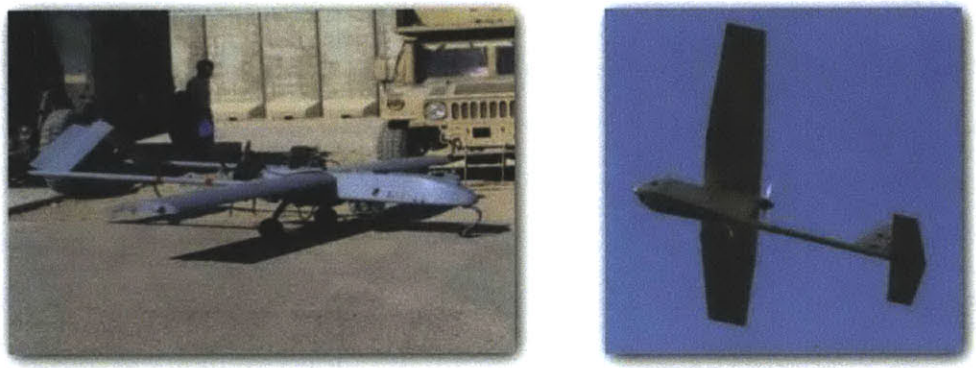
Figure 3 - Shadow TUAS Components. Although equipped with four aircraft, only 1 is typically flown at any given time. The Remote Video Terminals offer ground maneuver forces direct real time feed from the Shadow TUAS.

2.2.3.2 Raven RQ-11B Unmanned Aerial System

As a man-portable, hand-launched system, the RQ-11B Raven system consists of three unmanned aircraft, hand controller, ground control unit, and a remote video terminal. Its capabilities include an EO or IR payload (unlike the Shadow TUAS, only one payload can be employed with each flight), range of 10 km, airspeed of 27-60 mph, 90 min endurance, and also the ability to laser illuminate with IR. Units at the Company level and below in Iraq and Afghanistan use the Raven system for missions like force protection, to provide security during key leader engagements, and for counter indirect fire and counter-IED missions. Units can also use the One Source Remote Video Terminal (OSRVT) to remotely monitor live feed from the Raven while on the ground. Although usage of the Raven system varies between units, a typical unit will employ the Raven system for approximately 5 hours a day, 3 to 4 times a week, each mission lasting less than 90 minutes. Due to manpower constraints, units typically only employ one Raven at any given time, although multiple Ravens can operate in the same area with an upgraded digital data link. If the unit desires a longer on station time, the turn around time from one Raven landing to another taking off takes only about a minute, the time required to insert a new battery and to carry out a quick post flight check on the system. The main components of the Raven are shown in Figure 4.

Figure 4 - Raven system components. The Raven can only be outfitted with one of the two (IR or EO) payloads at any given time.

Units must also be careful to alleviate any frequency or airspace conflicts prior to each mission to avoid unnecessary crashes. In the event of a crash, it is at the commander's discretion whether to try and retrieve the system and the associated secure communication device [31]. Figure 5 summarizes the characteristics of both BCT UASs and provides a quick glance at the differences between the two UASs described.



RQ-7B Shadow UAS

RQ-11B Raven UAS

Wing Span	14 ft
Air Vehicle Weight	375 lbs
Range	60 km radius or 125 km range
Airspeed	70-120 mph
Altitude	Up to 15000' (Typical 4000 – 8000' AGL)
Endurance	9 hours
Payload	- Electro-Optical (EO)
	- Infrared (IR) with Laser Illuminator

Wing Span	4.5 ft
Air Vehicle Weight	4 lbs
Range	10+ km range (LOS)
Airspeed	27-60 mph
Altitude	>300' AGL
Endurance	1.5 hours Lithium
Payload	- Electro-Optical (EO)
	- Infrared (IR) with Laser Illuminator

Figure 5 - Brigade Combat Team UAS Types. The Shadow TUAS as a Brigade level asset offers extended range and endurance to cover the entire Brigade area of operations.

2.2.4 UAS Handover Operations

As part of basic level training, each Shadow TUAS and Raven UAS operator must know how to perform handover of the UAS to other operators, in accordance with the training manuals FM 3-4-155 and TC 1-611. Handover often refers to passing off targets to another UAS or manned aircraft, but it can also involve passing off a UAS to another unit. Since a Ground Control Station can only communicate with and control one UAS at a time, the operator can place a controlled UAS in programmed flight and be free to acquire another UAS during a handover. Although rarely performed in conflict, handovers of UASs can extend the range of the UAS especially with one way support missions like route clearance for assured mobility units [31].

2.3 Intelligence, Surveillance, and Reconnaissance Synchronization

2.3.1 From Desired Effects to Specific Information Requirements

Unlike conventional warfare, the complexity of COIN requires a flexible, dynamic, and multifaceted learning approach. In order to effectively manage Intelligence Requirements (IRs), the BCT Commander promotes some of his multiple IRs into Priority Intelligence Requirements (PIRs). Each PIR and IR breaks down further into observable, quantifiable, Specific Information Requirements (SIRs) for collection assets to focus on. For example, one of the three aforementioned conditions for success according to the COIN doctrine included the population viewing the government as legitimate, and controlling social, political, economic, and security institutions that meet the needs of the population. A good IR that addresses this would be "Why are host nation security forces unable to secure the area of operations?" The SIR associated with this IR might be the number of intelligence tips received by the local police station from the population. An increasing number of intelligence tips would not only indicate improved security but also increased trust with the local police force, proving its legitimacy and reliability. A poor SIR would be the number of police on the station roster because this fails to make the connection with legitimacy or security. As evident from this example, the BCT Commander along with his key staff members must generate precise, observable, quantifiable, and meaningful IR's linked to the Commander's desired effects [19]. An example of an SIR addressed by the Shadow TUAS might require an observation of the number of mosque and house vandalism events occurring during a sectarian conflict. As indicators, the Shadow TUAS can observe mosque explosions or haphazard roadblocks along roads to protect divided neighborhoods. This can provide the Commander with increased situational awareness and an opportunity to coordinate joint patrols with the HN security forces to generate trust.

2.3.2 Effects Working Groups

Most deployed units executing COIN operations hold biweekly Effects Working Groups (EWG) to discuss the BCT Commander's desired effects and revise PIR's along with their associated SIRs [16]. The staff members that attend these meetings typically include representatives for the following functional areas:

Functional area	**Duty Description**
S9 Civil Affairs	Identify critical requirements needed by local citizens, acts as liaison between population and Coalition
Surgeon	Handles coordination with local medical services and facility evaluations
Public Affairs and Information Operations	Works with attached PSYOPs (Psychological Operations) teams to execute effective information
Engineers	Works with Host Nation in building structures, barriers, and civil works program
Security	Responsible for coordination with local Police, Army, and other Security forces for training and recruitment
Economics	Works with the local Chamber of Commerce and oversees loans program
Governance	Works with local government organizations for reform and reconciliation
Essential Services	Handles contracting for local services including but not limited to sewage, potable water, power, trash, and
Judge Advocate General	Coordinates with local judicial department to enforce Rule of Law
S2 Intelligence	Responsible for acquiring, analyzing, disseminating intelligence. Plans collection operations with ISR
S3 Operations	Plans, controls, and executes mission operations as directed by the Commander

Table 1 - Staff Duty Descriptions. Each of these functional areas directly or indirectly support the 5 LLOs described in section 2.1.1

At these EWG meetings, each of the staff members discuss current and future operations, nested under the Commander's PIRs and IRs, and work to synchronize and deconflict efforts across the area of operations. For example, a plan to construct barriers surrounding a village to protect them from insurgents may directly conflict and cripple efforts to improve water or trash collection. Additionally, collective opportunities might arise like allowing the local Police Force to pass out information flyers to local residents. The EWG staff members ultimately assign priorities and produce a campaign plan for approval by the Commander in line with his PIRs and IRs. One of the primary outcomes of the Effects Working Group is a prioritized tasking of collection assets to answer SIRs. The S2 Collection Manager along with approval of the S3 Operations Officer handles the Intelligence, Surveillance, and Reconnaissance (ISR) Synchronization of assets to allocate tasks to sensors.

2.3.3 S2 Collection Manager Responsibilities

According to FM 2-01, the S2 Collection Manager (CM) takes the lead in performing ISR Synchronization, which includes analyzing information requirements and intelligence gaps, evaluating available internal and external assets, determining gaps in the use of those assets, recommending ISR assets to collect on the SIRs, and submitting Requests for Information (RFIs) for adjacent and higher collection support [7]. When evaluating collection assets, the S2 Collection Manager takes into account a myriad of factors like time constraints denoted by Earliest Time Information of Value (ETIOV) and Latest Time Information of Value (LTIOV), availability, and capability including performance history. Additionally, the S2 Collection Manager, with the approval of the S3 Operations Officer, must always strive to achieve balance when allocating assets, using redundancy when required to increase the probability of collection success, mixing different types of assets to achieve holistic understanding, and cueing to direct other assets to confirm or deny potential answers to SIRs. Key tasks during the ISR Synchronization process include the following six continuous activities shown in Figure 6:

- Develop requirements by converting SIRs into ISR tasks that tailor the reporting criteria to the collection capabilities of tasked assets
- Develop the ISR Synchronization Plan by attributing SIRs to specific collection assets based on capabilities and limitations. This includes forwarding SIRs that cannot be answered by organic assets to higher or lateral organizations as Requests for Information (RFIs)
- Support ISR integration with the BCT's operations
- Manage dissemination of information
- Assess the effectiveness of the ISR effort, identify gaps, and redirect
- Update the ISR Synchronization Plan as PIRs are answered and new requirements arise [8].

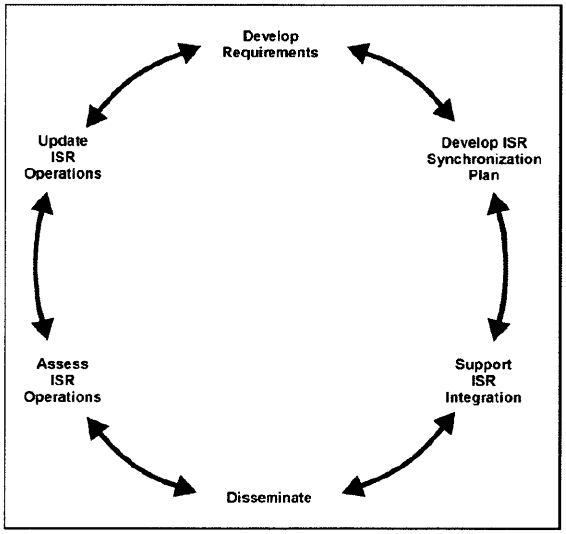


Figure 6 - ISR Synchronization Activities. As a continuous process, the S2 Collection Manager must always adapt to changing requirements and gaps.

In developing the ISR plan, the S2 Collection Manager must constantly balance between trying to fulfill all of the subordinate units' requests while addressing the BCT level SIRs resulting from the EWG meeting. Additionally, the S2 CM must also address any ISR taskings from higher echelons. For instance, for a BCT with three maneuver units, a realistic scenario might involve the S2 CM trying to apportion UAS time to support a raid for one unit, counter-IED for another, and counter-indirect fire for a third while also allocating time to cover a Division level main supply route.

To visualize where the ISR Synchronization would fit in COIN, Figure 7 depicts how the ISR Synchronization process stems from the SIRs produced by the Effects Working Group, part of the 'Threat, Environment, and Civil Consideration' [10]. This results in a closed loop process in which answers to Intelligence Requirements from ISR assets helps to revise PIRs and IRs and increase the overall situational awareness for the Commander.

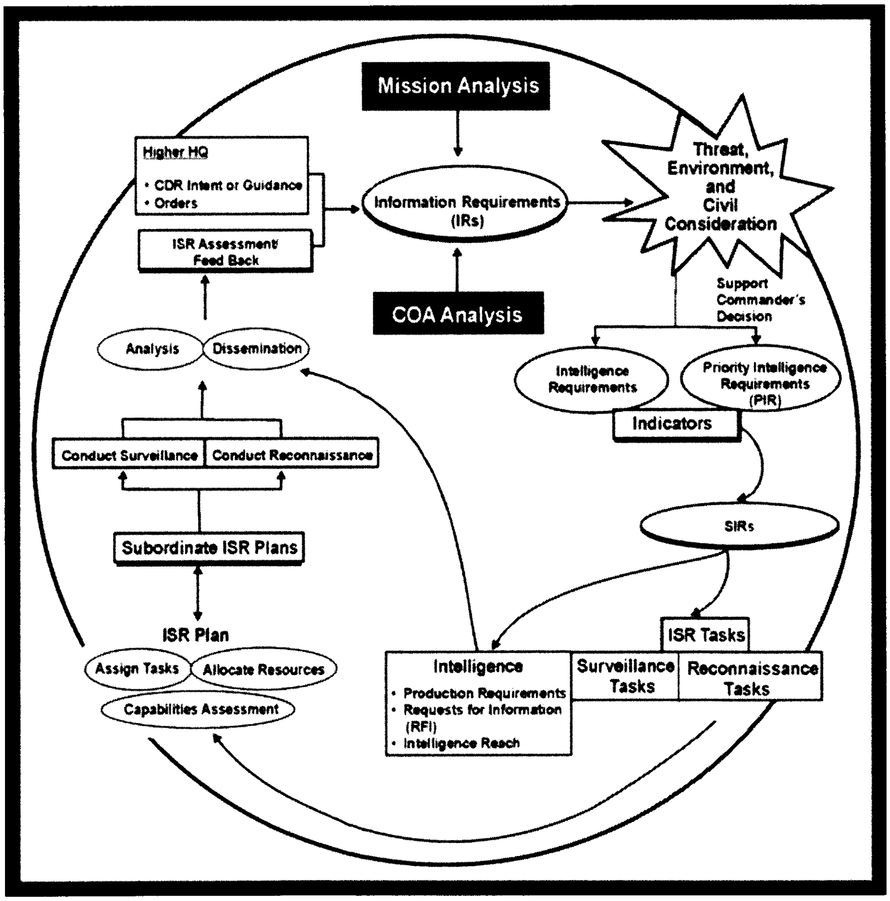
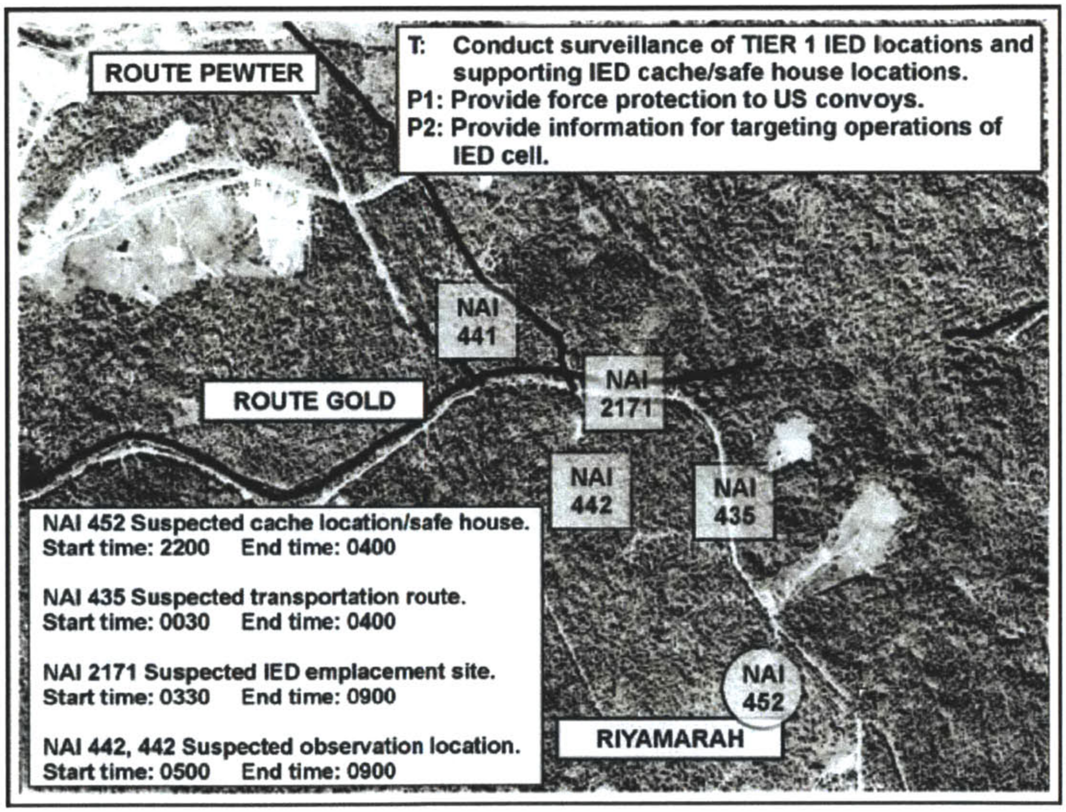


Figure 7 - ISR Synchronization Process. The ISR Synchronization makes up the bottom half of this flowchart. The process starts upon receipt of the SIRs and ultimately provides feedback for the revision of IRs.

In order to streamline ISR Synchronization, the S2 Collection Manager uses three tools to assist in dissemination of information and analysis: the ISR Synchronization Matrix (ISM), the ISR overlay, and the requirements matrix. Figure 8 is an example of a Battalion level ISR overlay for the task of conducting surveillance of a Tier 1 Improvised Explosive Device (IED) location, defined by each unit as a significant number of IED events in a 1-kilometer radius over a 30-day period. Labeled Named Areas of Interest (NAIs) identify where the insurgents' IED related activity might take place in both time and space.



T: Conduct surveillance of TIER 1 IED locations and supporting IED cache/safe house locations.
P1: Provide force protection to US convoys.
P2: Provide information for targeting operations of IED cell.

NAI 452 Suspected cache location/safe house.
Start time: 2200 End time: 0400

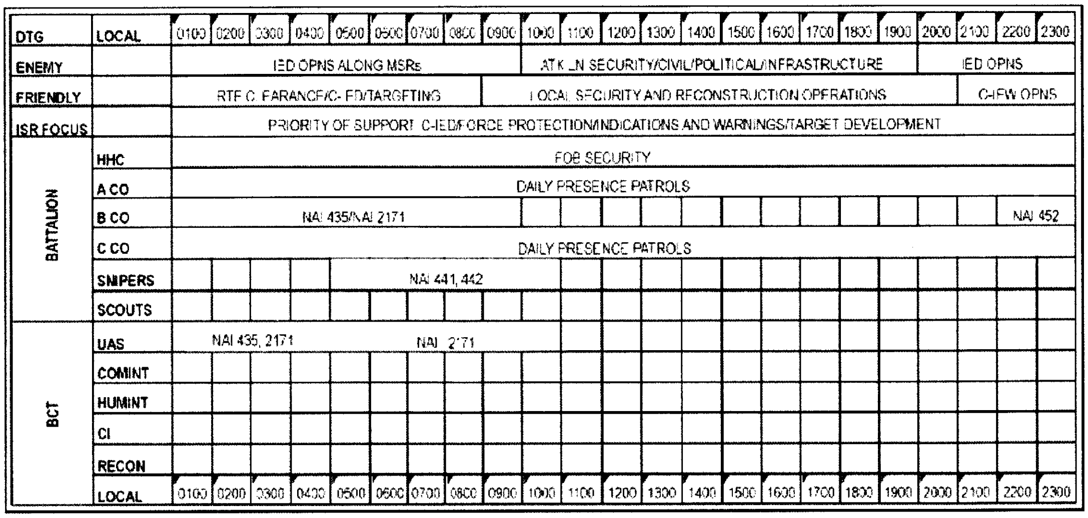
NAI 435 Suspected transportation route.
Start time: 0030 End time: 0400

NAI 2171 Suspected IED emplacement site.
Start time: 0330 End time: 0900

NAI 442, 442 Suspected observation location.
Start time: 0500 End time: 0900

Figure 8 - ISR Overlay. In this simplified example, each NAI supports the task and purpose which are nested under the commander's SIR.

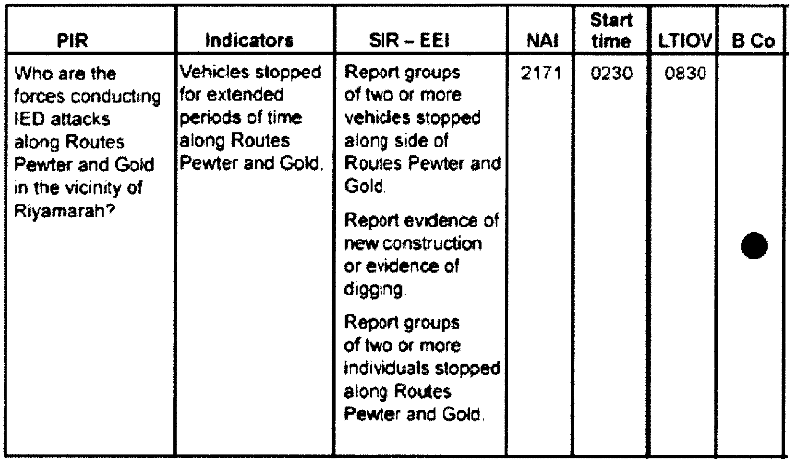
Figure 9 shows an example of a Battalion level ISR Synchronization Matrix that units might use in theater showing the timeline for collection for different assets in specific NAIs.



DTG	LOCAL	0100	0200	0300	0400	0500	0600	0700	0800	0900	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200	2300
ENEMY		IED OPNS ALONG MSRs									ATK ON SECURITY/CIVIL/POLITICAL/INFRASTRUCTURE										IED OPNS			
FRIENDLY		RTE CLEARANCE/C-IED/TARGETING								LOCAL SECURITY AND RECONSTRUCTION OPERATIONS												C-IEW OPNS		
ISR FOCUS		PRIORITY OF SUPPORT C-IED/FORCE PROTECTION/INDICATIONS AND WARNINGS/TARGET DEVELOPMENT																						
BATTALION	HHC	FOB SECURITY																						
	A CO	DAILY PRESENCE PATROLS																						
	B CO	NAI 435/NAI 2171																					NAI 452	
	C CO	DAILY PRESENCE PATROLS																						
	SNIPERS					NAI 441, 442																		
	SCOUTS																							
BCT	UAS	NAI 435, 2171					NAI 2171																	
	COMINT																							
	HUMINT																							
	CI																							
	RECON																							
	LOCAL	0100	0200	0300	0400	0500	0600	0700	0800	0900	1000	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200	2300

Figure 9 - ISR Synchronization Matrix. In this example, the Brigade Shadow TUAS support to this Battalion scheduled from 0100 to 1100 hours covers NAIs 435 and 2171.

Figure 10 shows a small portion of a requirements matrix that associates each collection asset with indicators and SIRs. Although in practice units employ different versions of these tools, they all provide each collection asset with the critical information to execute its mission properly. As depicted, each SIR falls under the umbrella Priority Intelligence Requirement addressing IED attacks in sector [34]. The indicators assist the Collection Manager in deciding on the optimal available sensor to address the SIR. In this example, the indicators are vehicles stopped alongside the road for extended periods of time. Another physical indicator example for an LLO related SIR would be the number of Soldiers observed checking vehicles to address the question "How many Host Nation Soldiers are manning checkpoint 1 from time A to B?" under the PIR "Why can't host nation security forces secure the area of operation?"



PIR	Indicators	SIR – EEI	NAI	Start time	LTIOV	B Co
Who are the forces conducting IED attacks along Routes Pewter and Gold in the vicinity of Riyamarah?	Vehicles stopped for extended periods of time along Routes Pewter and Gold.	Report groups of two or more vehicles stopped along side of Routes Pewter and Gold. Report evidence of new construction or evidence of digging. Report groups of two or more individuals stopped along Routes Pewter and Gold.	2171	0230	0830	●

Figure 10 - Requirements Matrix. This tool helps the S2 collection manager to tie each NAI to the commander's SIR and ultimately to its associated PIR.

2.3.4 Ad hoc and Dynamic Retasking Before and During Mission Execution

During or before execution of the ISR plan, the BCT S3 Operations Officer and BCT Commander handle any deviation in allocation or apportionment of assets based on operational priorities. Whereas dynamic retasking relates to changes in the mission of a collection asset while in the execution phase, ad hoc retasking relates to changes in the mission after planning but before the execution phase [34]. External factors like unforeseen weather constraints may prevent a UAS from supporting a mission as planned; in this case, the S3 may retask a ground asset to support the mission. Examples of dynamic retasking targets for the UAS include:

1) Troops in Contact (TIC) and Battle Damage Assessment (BDA);
2) Time Sensitive Targets (TST); and
3) Reconnaissance to confirm or deny critical intelligence.

With retasking collection assets, the S3 must always consider the priority level of the new requirement relative to remaining unsatisfied requirements. For example, the S3 may dynamically retask a Shadow TUAS executing a Counter-IED route clearance mission to provide close support to troops receiving contact from the enemy. Due to perturbations to the original schedule because of these dynamic retaskings, units often experience significant loiter time while waiting for the next scheduled target window. Often units will struggle to find additional targets for a UAS given this extra time; this supports the need for secondary target sets that might incorporate the LLO targets stemming from the Effects Working Group. Some UAS pilots and mission commanders resort to nominating secondary targets on their own after soliciting the supported unit to no avail, which despite the efforts of the UAS unit, circumvents the entire ISR Synchronization process.

2.4 UAS Role: Latter Stages of Coin

As the COIN phase approaches the end of the aforementioned Inpatient care and Recovery stage, the focus shifts from military primacy and kinetic operations to multidimensional non-kinetic operations. BCT Commanders might allocate ISR assets to match the shift in focus. For example, if the majority of PIRs address non-lethal targets, the missions of the BCT UAS assets should largely reflect this. The BCT level UAS's Quick Reaction Force (QRF) role in supporting Troops in Contact (TIC) and protecting Soldiers' lives will remain paramount; however, the majority of its other missions should properly align with SIRs developed from the Effects Working Group.

Nothing can replace the value of human intelligence, especially when it comes to COIN. Yet the Imagery Intelligence (IMINT) collected by the UAS not only enhances the situational understanding; it also provides the Commander with a unique unadulterated view of the five LLOs and current state with respect to each. In irregular warfare operations like COIN, the traditional mix of ISR assets used in conventional operations may not satisfy the commander's information requirements. Irregular warfare operations sometimes require unconventional thinking in terms of ISR planning [7].

2.4.1 Potential LLO Secondary Targets and Indicators for the UAS in COIN

According to FM 2-19.4, indicators are "any positive or negative clue that points toward threat activities, capabilities, vulnerabilities, or intentions" [8]. Used in the context of the COIN environment, one can replace the word 'threat' in this definition with Host Nation. In order to properly allocate tasks to sensors, the collection manager must first decompose the observables and then prioritize those that provide the best indicators given the sensor and its capabilities. Military Transition Teams (MiTTs), Police Transition Teams (PTTs), Provincial Reconstruction Teams (PRTs), Civil Affairs Teams, and maneuver combat patrols address most or all of these observables; however, only the UAS provides an honest assessment on progress in the absence of Coalition Force presence. The indicators observed by UASs can help analyze the patterns of life, defined as the rhythm of human activity, for a specific location. Table 2 shows a sample of potential LLO secondary targets, its observables, and indicators with which the Collection Manager might task the UAS to observe in the latter stages of COIN concerning each line of operation:

LLO	LLO Targets	Observables	Indicators for UAS
1) Conduct combat operations/ civil security operations	Independent Host Nation security operations	Independent raid operations, detainee operations, security posture, planning meetings	Host Nation police directing traffic
	Security status along borders	Personnel and vehicle checks conducted, guards present with appropriate uniforms and identification, fed, and paid	Troop levels, burning trucks, Soldiers checking vehicles and personnel
2) Train and employ HN security forces	Checkpoint operations	Personnel and vehicle checks, shift change, Soldiers uniformed, fed, and paid. No corrupt activities	Soldiers checking vehicles and personnel
	Unit Training and logistics	Unit readiness training, qualification range, physical training, classes, documentation, reports, logistics	Soldiers on firing range, vehicle maintenance activity, physical training, formations

	Humanitarian assistance operations	Organization of personnel and vehicles, reports and proper documentation	Orderly food or blanket distribution
3) Assess, Establish or Restore essential services	Trash pickup operations	Schedule, trash truck operation, trash on streets, paid workers	Trash along streets, trash truck on schedule
	Electricity	Voltage reports, working lights, plant workers, plans to expand powerlines	Powerlines, working traffic lights, night lights
	Security status along oil and water pipelines	Measurements of flow reports, structure and maintenance of pipelines, fires	Burning fires, security presence, broken pipes
	School status	Children and teachers in class, availability of resources	School population during start and end
	Fuel stations	Lines at Fuel Stations, black market sales	Fuel station lines, presence of individuals with fuel cans on sides of roads
4) Support development of better governance	Polling centers during elections	Orderly polling centers, security checks, proper documentation	Amount of people at polling centers, security presence, checking personnel
	Churches and religious services	Number of people attending, parked cars, condition of building	Vandalism or destruction, attendance levels
	Infrastructure and civil project locations	Proper documentation, gradual progress, workers present, proper material used	Road maintenance crew, construction crew and equipment present
5) Support economical development	Farming activity	Legal farming activity, types of crops	Number of poppy fields using Infrared IR or EO sensor
	Commercial activity	Types of stores, traffic, social activity at night, pedestrian traffic, money exchange, types of goods available, prices	Vehicle traffic, Night social activity, market pedestrian traffic

Table 2 - LLO Observables and Indicators. Given the limits of the UAS, required observations may not be satisfied. For example, to assess host nation security force checkpoint operations, the UAS can only observe soldiers checking vehicles and personnel. Bribing or corruption activities would be difficult to assess.

Each LLO requires different time windows for observation. For example, the optimal time for the UAS to observe school activity might occur twice a day, before and after classes, whereas trash pickup operations occur once a week following a preset schedule. Military Transition Teams can benefit from the UAS observing independent Host Nation security force operations during a preplanned raid only during the hours of operation. A spate of violence against minority Christian religious groups might require observation of churches according to congregational meeting schedules. Additionally, some of these LLO targets might benefit from a dual visit from different UAS types. For example, to facilitate effective change or anomaly detection, two different views by different UASs might offer the right amount of data to identify an IED wire along a road. Another application might involve two looks at a road construction project to verify timely completion of work. Most of the LLO targets listed require only a cursory look such as the requirement to observe queues forming at gas stations or commercial activity observations.

2.4.2 Primary UAS Missions in COIN

The following are examples of primary missions that the S2 Collection Manager might schedule UASs to execute in the latter stages of COIN based on subordinate requests and ISR taskings from higher echelons:

1) Information Operations. The UAS will fulfill an increasing future role in disseminating leaflets and broadcasting messages in supporting Information Operations. Currently the U.S. Army typically releases devices like the M129E1/E2 Psychological Operations Leaflet 'Bomb' from various manned aerial assets to disperse 60,000 to 80,000 leaflets at a time. In the latter stages of COIN, especially with significant withdrawal of forces, Coalition Forces must continue to promote the Host Nation's viability.
2) Counter-Improvised Explosive Device (C-IED). Recently outfitted with highly advanced change detection capabilities and teamed with manned assets, the rate of advance for UASs dramatically improved in recent years. With the decreased amount of route clearance patrols in sector,

Coalition Forces must maintain the ability to clear routes not only for themselves but more importantly, for Host Nation security forces.

3) Counter-Indirect Fire (C-IDF). As CF withdraw to fewer major bases, indirect fire attacks could evolve as the preferred attack mode for insurgents.
4) Close Air Support to ground maneuver forces. Joint cordon and search operations with Host Nation security forces to kill or capture High Value Targets (HVT) will continue.
5) Signals Intelligence (SIGINT) targeting and communications relay. With advancements in payload technology, the use of UASs to perform SIGINT roles will continue to expand.

2.5 Research Goals

2.5.1 Current Operations Framework

For most units, the ISR process at the Brigade level does not include synchronization between Battalion and Company level operated Ravens. The collective ISR Synchronization matrix may be accessible by every level, but Battalions and Companies typically do not attempt to share targets across well-defined unit boundaries. In fact, the design of the Raven system as a tactical level UAS encourages this practice of decentralized planning. Most Battalions and Companies that use the Raven system develop plans that concern local targets and disseminate imagery products accordingly to subordinate units. Concerning the Raven UAS system, coordination rarely occurs between Battalions or between Companies.

2.5.2 Shifting Paradigms

A shift in this paradigm by encouraging synchronization between Raven units would not take too much additional effort other than a slight adjustment of battle rhythms because all units already maintain connectivity with each other. The most important adjustment would involve deconfliction of airspace. The potential advantages of blurring unit boundaries could prove significant especially with limited resources during the second stage of COIN as units continue to withdraw from the Host Nation.

Based on the limited range and endurance of the Raven system, this thesis will research the advantages of intra-Brigade Raven target synchronization across Battalions and Companies. This supports the notion of effective COIN operations according to FM 3-24 to provide the lowest echelons increased capability.

2.5.3 Research Objectives

Given a unit's primary UAS missions and the additional LLO targets available at any moment with different time windows, the most optimal flight path in a COIN environment would attempt to address the maximum amount of priority LLO SIRs along its route. Recognizing that Commanders achieve success in COIN by way of information supremacy, this research focuses on maximizing the situational awareness during the second stage of COIN through the use of reward points gained from addressing LLO targets while enroute to primary missions. The principal goals of this thesis include:

1) Formulate a tractable concept that closely resembles reality and handles a heterogenous set of targets;
2) Show the significant gains in target coverage attained by avoiding stovepipe plans and allowing target and base sharing with respect to Raven UASs; and
3) Create a multiple sortie plan that will provide maximum value to the Brigade Commander within a timeframe that meets practical operational constraints.

3 Model Development

With the goals outlined in the previous chapter, this chapter provides the framework for the formulation of our Collective UAS Planning Problem (CUPP) and a review of literature.

3.1.1 Structure of Problem

Similar to the Vehicle Routing Problem discussed, our concepts will make use of a network representation in the form of a graph with defined nodes and undirected arcs, G(N,A), as well as reward values, time windows, and observation times for each target node. Realistically, each target requires specific requirements defined by time, location, and frequency of observations. For example, the ability to recognize a glint off of a target might require multiple passes at certain payload angles. As another example, an observation requirement along an oil pipeline or border might require an exact route along certain waypoints. For the purposes of this thesis and in large part due to the scale of the problem, each of these particular routing requirements will be aggregated within a node or in the case of the pipeline, nodes at critical waypoints.

3.1.2 Types of Targets

Our model will reflect three different types of targets: Primary mission targets, secondary LLO single look targets, and secondary LLO dual look targets. Primary mission targets require the most amount of time and encompass the COIN tasks described in Section 2.4.2 like Counter-IED and Information Operations. Most of the LLO secondary targets will only require a cursory look involving observations of market activity or local police checkpoints for example. Whereas the first two types of targets can be visited only once, dual look targets can be visited twice but only with different UASs. As our earlier change detection example illustrates, two different views at different times with different payload capabilities might offer the right amount of data to identify an IED wire along a road. Figure 11 illustrates an example of a graph of 2 UASs and 5 targets along with example routes.

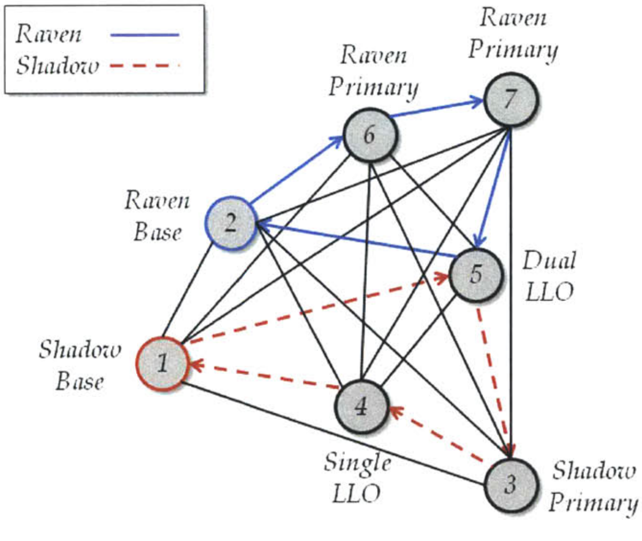


Figure 11 - Example UAS Routes. The Shadow and Raven UASs observe the dual look LLO target along with its primary required targets. The single look LLO is satisfied by the Shadow TUAS.

3.1.3 Assumptions

All flight paths between nodes will be direct under the assumptions of the triangular inequality, given the locations in a 3-dimensional Euclidean space. Therefore the distance from i to j will equal the distance from j to i and $distance_{ij} \leq distance_{ikj}$ for any additional point k. Additionally, despite the similarities in capability of the EO and IR sensors on both the Shadow TUAS and Raven UAS, we will assume that each individual UAS offers different payload capabilities. Although imagery analysts typically use the National Imagery Interpretability Rating Scale (NIIRS) from 0 – 9 to quantify image quality, our model will primarily be concerned with differentiating between the two UAS types. Therefore, we assume that the Effects Working Group (EWG) will appoint higher values for Shadow TUAS observations versus those of the Ravens based on payload weight allowance and associated image quality.

Our model assumes that the EWG determines the values for each LLO target node based on type. The EWG will always appoint higher values for priority targets based on the Commander's inputs. For instance, a cursory-look target like a marketplace observation requirement might receive a single value lower than a dual look target like a civil project, for which the value will be greater if observed independently by two different UASs. We assume that successful observation of a dual look target will result in a higher value than the sum of the independent single look values because of the additional benefit gained from different sensors.

Terrain factors like mountains, vegetation, or urban terrain can impact not only visibility but also how the UAS operates in terms of mission duration, altitude, and observation angles. Additionally, dense clouds or severe weather can also impair UAS missions because of visibility and equipment limitations. Although terrain and weather can severely impede UAS operations, our model will not account for them for simplification purposes.

Because the inclusion of a diverse set of targets can easily become complex, we set our research bounds to ensure that our focus remains concise. The research bounds are based on operational scenarios and despite the simplifications, remain close to actuality:

1. The maximum number of UASs explored will be five, remaining within the bounds of a Brigade Combat Team's steady state operating environment;
2. The number of different types of UASs used will be two, with one Shadow TUAS and up to 4 Raven UASs;
3. The maximum number of target nodes will be 50; and
4. Despite opportunities for split site operations, we assume that the Shadow TUAS launch and recovery site is collocated with the Ground Control Station.

3.1.4 Inputs

Our inputs to our model regarding the targets include the following:

1. A non-negative benefit value for all observations including both single and dual look targets.
2. The Earliest Time Information of Value (ETIOV) and Latest Time Information of Value (LTIOV) for each target. If a UAS arrives before the ETIOV, it must wait until the ETIOV to execute the given observation task as illustrated in Figure 12. If a UAS arrives after the LTIOV or at a time that does not allow sufficient observation time, then execution of the task cannot occur. Otherwise, if the UAS arrives between the ETIOV and LTIOV with a sufficient observation time window, execution of the given observation task can begin upon arrival. While LLO target time windows can span the length of the entire horizon, units will give primary mission target nodes a narrower focus in terms of time.

3. The observation time requested at each target. Primary mission target nodes will generally require longer observation times than LLO target nodes. In order to gain the value associated with a target, the UAS must remain at the target node until it fulfills the observation time requirement.
4. The altitude required at each target for noise considerations, desired effects, and image quality. For example, lower altitudes might be requested for deterrence of rocket attacks because of the clearly audible presence of the Shadow TUAS at certain altitudes. In the lexicon of close air support, this is similar to what is defined as a 'show of presence,' a nonlethal display to both reassure ground units or to deter enemy units.

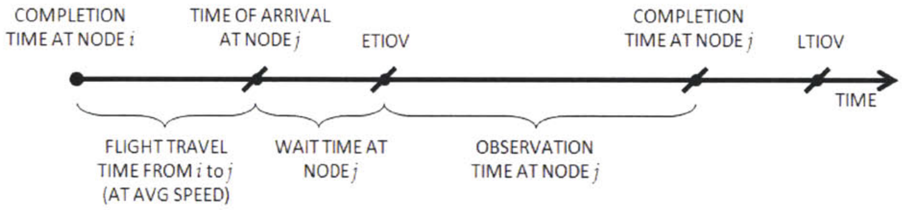


Figure 12 - Timeline Between Nodes.

Figure 13 summarizes the above in a simple graph consisting of two nodes i and j connected by arc ij.

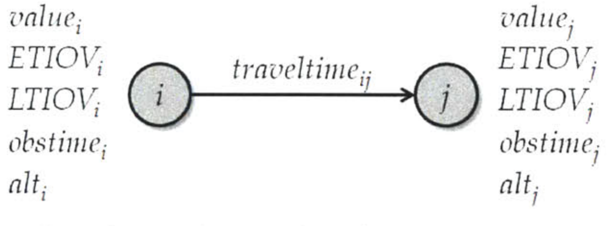


Figure 13 - Graph Node Notation. Each node is characterized by five inputs.

The inputs to our model regarding the Shadow and Raven UAS types include the following:

1. The endurance limit for each UAS according to Figure 5.

2. The average speed for each UAS
3. Average descent rate and climb rate for each UAS
4. Base locations for both the Shadow and Raven UAS
5. Maximum and minimum operating altitudes for both UASs
6. Earliest time each UAS is available for missions based on maintenance requirements

Based on the speed of the UAS and altitude required at each target node, the travel time between nodes i and j can be calculated with the following equation:

$$traveltime_{ij} = \frac{xy\ distance}{UAS\ speed} + \frac{altitude\ change\ (z\ distance)}{UAS\ sinkrate/climbrate}$$

3.1.5 Outputs

The outputs of our model include the following:

1. The maximum objective function value encompassing the aggregated total value achieved by target observations
2. The routes identified by node order for each UAS
3. The schedule including the departure and arrival times at each node for each UAS
4. The total mission time required for each UAS
5. The total wait time or idle time spent at each node

3.2 Literature Review and Problem Classification

This section presents a summary of literature written on similar and related problems as the one posed in this thesis.

3.2.1 Traveling Salesman Problem

Because the CUPP can be classified as a node covering problem, the Traveling Salesman Problem (TSP) provides a fundamental basis for the development of our concept. Aptly named after a salesman trying to make efficient use of time and resources, the TSP aims to "find the minimum distance route that begins at a given node of a network, visits all of the members of a specified set of nodes on the network at least once, and returns eventually to the initial node" [20]. Typical TSP problems specify that each node must be visited exactly once in a network that is completely connected, meaning that each node can be reached from any other node without having to go through other nodes in the set. Furthermore, typical TSP problems also satisfy the triangular inequality, such that the lengths l of direct links between any three points i, j, and k satisfies $l(i,j) \leq l(i,k) + l(k,j)$.

Solving for the exact solution of a TSP problem, especially with a large-scale, operationally sized problem involving hundreds of nodes, can take days or even longer to solve even with the most recent advances in technology. Given a problem with $n-1$ nodes to be visited by a single vehicle, there are $(n-1)!$ possible solutions to explore. To date, although exact algorithms were developed over the years to solve problems with up to 48,000 nodes, none of the algorithms are efficient in terms of being solved in polynomial time. The TSP belongs to a special class of difficult combinatorial problems called NP-hard problems, for which no efficient algorithm may ever be found to solve it optimally. One of the existing exact solution algorithms based on dynamic programming is the Held-Karp Algorithm, which solves the problem in time $O(n^2 2^n)$, varying exponentially with the number of nodes, n [17].

For the TSP, well performing heuristic algorithms exist that provide approximate solutions with minimal computation costs. Most of these heuristics make use of hybrid heuristics, making use of both construction heuristics that build solutions from scratch as well as improvement heuristics that try to improve upon a given solution, usually by making small changes. To date, the Christofides heuristic provides the best worst case algorithm with $\frac{3}{2}L(TSP)$ as an upper bound for the tour length, where $L(TSP)$ denotes the length of the optimum traveling salesman tour.

Well known construction heuristics include the nearest neighbor heuristic, a greedy algorithm that iteratively adds the closest neighbors to the tour. This heuristic has a poor worst case performance with $\frac{L(NN)}{L(TSP)} \leq \frac{1}{2}\lceil \log_2 n \rceil + \frac{1}{2}$. For example, given 128 nodes, the solution would be $\frac{1}{2}(7) + \frac{1}{2}$ or four times greater than optimal. Other known construction heuristics include random insertion with a worst case performance of $\lceil \log_2 n \rceil + 1$, farthest insertion, nearest insertion, and cheapest insertion with the random and farthest heuristics performing better on average because they are not greedy algorithms. The nearest and cheapest heuristics have worst case performances of 2 times greater than optimal.

Most improvement heuristics search the neighborhood $N(T)$ of a tour T. Well known improvement heuristics include the 2 exchange heuristic, where $N(T)$ consists of all of the tours that can be found from T by deleting two arcs and inserting two. Since there are $\binom{n}{2}$ ways of choosing these two arcs, this heuristic is proportional to $O(n^2)$. Other improvement heuristics include the 3 exchange or 3-opt, which is proportional to $O(n^3)$ and extensions to the k-opt, where k is the number of arcs exchanged.

Although the TSP provides the initial groundwork for this thesis, the unique constraints and characteristics posed by our problem requires further review of literature.

3.2.2 Vehicle Routing Problem

Largely applied to supply chain design, the Vehicle Routing Problem (VRP) involves finding a set of routes, starting and ending at a base or depot, that together serve a set of customers. Each customer's demand is known and each vehicle can only service as many customers as its capacity permits. In addition, maximum route time constraints must be satisfied. The typical objective of a VRP is to minimize the total travel distance covered by the entire fleet or the number of vehicles used, or a combination of the two. When more than one vehicle exists, the VRP reduces to a m-TSP problem where m denotes the number of vehicles to be used. The additional constraints imply that the VRP will be expected to be more difficult to solve optimally than a TSP.

3.2.3 Vehicle Routing Problem with Time Windows

Vehicle Routing Problem with Time Windows (VRPTW) is a generalization of the VRP where the service for any customer starts within a set time interval called a time window. Time windows can either be termed 'soft' if they are non-binding and typically incur a set penalty or 'hard' if they must be obeyed. In the latter case, vehicles cannot arrive after the time window closes and if a vehicle arrives too early, it must wait until the time window starts to serve a customer [21].

3.2.4 Generalizations to the TSP and VRP

A common characteristic of both the TSP and VRP is the requirement to service every customer, without assigning values. Generalizations to these problems involve selecting customers based on a certain value or profit gained for visiting each customer. When a single vehicle is involved, these types are problems are called Traveling Salesman Problems with Profits (TSPs with Profits) [6]. By changing the objective function, additional problem characterizations emerge. For example, the objective function might involve the maximization of the collected total profit (Orienteering Problem), the minimization of the total traveling cost (Prize-Collecting TSP), or the optimization of a combination of both (Profitable Tour Problem) [1]. Of these, the Orienteering Problem (OP) comes closest to the objective function for the CUPP.

3.2.5 Orienteering Problem

Based on the sport of orienteering, a competition between individuals scored according to the number of pre-determined marker locations reached within a specified time limit, the Orienteering Problem (OP) is also known as the Selective Traveling Salesman Problem (STSP) [1]. Originally investigated by Tsiligirides in 1984, the OP involves a single vehicle that must reach destination points prior to time T_{max}. In addition to the start and end points, the vehicle can visit a set of locations with designated point values. The objective of the OP is to determine the optimal route to maximize collected value before T_{max}. OP problems were determined to be NP-hard by Golden et al. in 1987, and therefore most of the literature focuses on heuristic approaches [15].

3.2.6 Team Orienteering Problem with Time Windows

As a generalization to the OP, the Team Orienteering Problem (TOP) and its variant with time windows (TOPTW) involves multiple tours, based on orienteering teams working together to attain the highest score. The extension of OP to multiple tours was introduced under the name TOP by Chao et al [4]. Unlike the standard TSP where arcs must not intersect for an optimal solution, strict time windows with the TOPTW allow for intersections of arcs between nodes. Righini and Salani [28] used bi-directional dynamic programming to solve the Orienteering Problem with Time Windows (OPTW) optimally.

Garcia et al. [13] solve a Multi-Constrained Team Orienteering Problem with Time Windows (MCTOPTW) to generate personalized tourist routes, providing a useful reference for the CUPP. Given a tourist's interests and constraints with budget and time, the objective of the MCTOPTW is to maximize the collective score without violating the constraints. Each location not only has a score associated with it, but also additional attributes like visit duration and entrance fee. They also formulate the MCTOPTW as an integer problem with 10 sets of constraints.

3.2.7 Aircraft Routing Problems

Aircraft Routing Problems assign individual aircraft to flight legs while satisfying maintenance requirements. The objective is to find a feasible assignment of individual aircraft to scheduled flights so that each flight is covered exactly once, maintenance requirements are satisfied, and the flow balance of aircraft is maintained for the number of aircraft available. Barnhart et al. [2] define strings in an aircraft maintenance routing problem as sequences of flights beginning and ending at a maintenance station that satisfy flow balance.

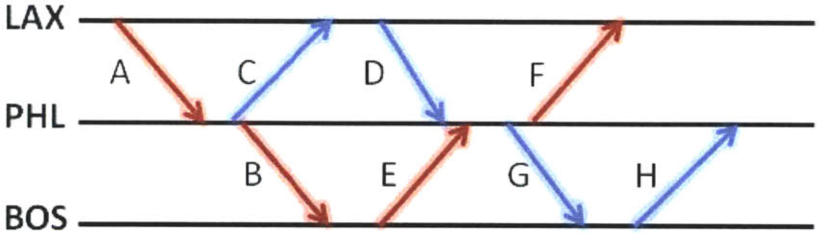


Figure 14 - String Example. String 1 (a-b-e-f) starts and ends in LA with maintenance stops in Philadelphia, Boston, and Philadelphia again, while String 2 (c-d-g-h) starts and ends in Philadelphia with maintenance stops in LA, Philadelphia, and Boston. Each string satisfies the maintenance constraints.

Because the problem posed by our thesis addresses multiple sorties within a planning horizon, aircraft maintenance routing problems prove relevant to our formulation.

3.3 Review of Heuristics

Due to the high computational costs of all of the combinatorial optimization problems reviewed thus far, we explore well known heuristics to apply to our problem in order to efficiently reach approximate solutions.

3.3.1 Iterative Local Search

Regarded by many as the most popular heuristic, Local Search (LS) uses an iterative search procedure to improve a given initial solution by applying local modifications. Each iteration provides a move to an improved feasible solution until a local optimum is reached. The Local Search heuristic uses an insert step that tries to add new locations to a route by calculating a ratio that takes into account the location score and insertion time. The inability to search outside of this local optimum limits the quality of the solution.

The Iterated Local Search (ILS), categorized as a metaheuristic, iteratively builds solutions generated by LS methods. Because perturbations of the local search solutions are used to create new solutions, this metaheuristic provides an improvement over randomly repeating the same heuristic. The heuristic uses a technique called iterated local search with a random walk criterion, because it always continues the search from the current solution, and never reverts to the best solution found. A general ILS metaheuristic algorithm can be summarized as follows:

```
Iterated Local Search
$s_0$ = GenerateInitialSolution;
$s^*$ = LocalSearch ($s_0$);
while termination criteria NOT met do
    $s'$ = Perturbation ($s^*$);
    $s^{*'}$ = LocalSearch ($s'$);
    $s^*$ = AcceptanceCriterion ($s^*, s^{*'}$);
```

Vansteenwegen et al. [37] used the following ILS algorithm to solve the TOPTW efficiently, including a shake step to provide an escape from local optima. The shake step removes visits at least once and helps to expose the entire solution space. Here *n* denotes the number of locations and *m*, the number of routes.

```
Iterated Local Search
StartPosition = 1;
NumberToRemove = 1;
NumberOfTimesNoImprovement = 0;
while NumberOfTimesNoImprovement < 150 do
    Insert local search;
    if Solution better than BestFound then
        BestFound = Solution;
        NumberToRemove = 1;
        NumberOfTimesNoImprovement = 0;
    else
        NumberOfTimesNoImprovement +1;
    end
    Shake Solution (NumberToRemove, StartPosition);
    StartPosition = StartPosition + NumberToRemove;
    NumberToRemove +1;
    if StartPosition > Size of smallest route then
        StartPosition = StartPosition - Size of smallest route;
    end
    if NumberToRemove == n/3m then
        NumberToRemove = 1;
    end
end
Return BestFound
```

3.3.2 Simulated Annealing

As another local search metaheuristic that allows an escape from local optima, Simulated Annealing allows hill climbing moves and moves that degrade the value of the objective function in order to find the global optimum. First described by Scott Kirkpatrick et al in 1983, the name comes from an analogy to the process of annealing in metallurgy, a technique that involves heating and cooling of a material to increase crystal size and reduce defects. The heat forces atoms to leave their original positions (local minimum) and wander randomly as the cooling process helps the atoms to locate lower energy configurations and be free of crystal defects with superior structural integrity. With each iteration the simulated annealing algorithm compares the incumbent solution and the new solution. Improving solutions are always accepted while a fraction of inferior solutions are accepted based on a temperature parameter, the probability of accepting inferior solutions.

Simulated Annealing starts with an initial solution $\omega \in \Omega$. One then generates a neighboring solution $\omega' \in N(\omega)$ either randomly or according to a pre-specified rule. The candidate solution's acceptance is based on the following probability where t_k is defined as the temperature parameter at iteration k [14].

$$P\{Accept\ \omega'\} = \begin{cases} \exp\left[\dfrac{-f(\omega') - f(\omega)}{t_k}\right], & if\ f(\omega') - f(\omega) > 0 \\ 1, & if\ f(\omega') - f(\omega) \leq 0 \end{cases}$$

A large initial temperature parameter typically decreases in close analogy to the cooling process of annealing as the algorithm progresses through its iterations. As the temperature decreases in this manner, the probability of reaching the global optimum reaches 1 as time approaches infinity.

3.3.3 Tabu Search

First proposed by Fred Glover in 1986, the Tabu Search (TS) metaheuristic became very popular after successful implementations with complex combinatorial problems. In order to overcome the limitations of the Local Search heuristic, TS extends the concept of LS by using short term memory to label potential solutions as taboo in order to prevent cycling and to avoid being ensnared by local optima. Tabus are stored in a tabu list and usually include only a limited amount of information instead of complete solutions. TS defines a 'move' as any change in the current solution and also defines the adjacent solution as a 'neighbor' of the current solution. During each iteration, TS searches through the neighboring solutions and picks the neighbor with the highest objective function value to become the new incumbent. The most common tabus record the last few moves applied to an initial solution and forbid the reverse moves. For example in the classic Vehicle Routing Problem, if a vehicle v_1 was moved from Route R_1 to Route R_2, the tabu would restrict v_1 from moving from Route R_2 to Route R_1 for a certain number of iterations n, where n is described as the tabu tenure. One potential drawback of tabus is that some tabus may end up denying quality moves that could lead to better solutions. A way to counter this is to use Aspiration criteria, an algorithmic tool to allow for the cancellation of tabus. A commonly used aspiration criterion allows a tabu move if it results in a better objective value than the current incumbent solution. The general template for TS is to minimize a function $f(S)$ using the most common 'best improvement' version, the version that chooses the best available move at each iteration.

Notation

S	Current solution
S^*	Best known solution
F^*	Value of S^*
$N(S)$	Neighborhood of S
$\widetilde{N}(S)$	Admissible subset of $N(S)$, non-tabu or allowed by aspiration criteria
T	Tabu list

```
Initialization
Choose an initial solution $S_0$
Set S← $S_0, f^* \leftarrow f(S_0),\ S^* \leftarrow S_0,\ T \leftarrow \emptyset$

Search
While termination criterion not satisfied, do
Select $S$ in $argmin_{S' \in \widetilde{N}(S)}[f(S')]$;
if $f(S) < f^*$, then set $f^* \leftarrow f(S),\ S^* \leftarrow S$;
Record tabu for the current move in T (delete oldest entry if necessary)
```

The most common termination criteria used with TS involve stopping after a fixed number of iterations, after some number of iterations without an improvement in the objective function value, or after the objective reaches a specified value. An additional way to improve the effectiveness of the search strategy includes search intensification, or to focus more of the search in the more promising areas of the search space. For example, with the Vehicle Routing Problem, one could restart the search from the best incumbent solution and fix the attractive components like an arc that is used consistently. Although TS is a proven algorithmic approach, its parameters that include neighborhood structure, aspiration conditions, form of tabu moves, size of the tabu list, and the termination criteria, require extensive calibration and systematic testing [26]. Tang and Miller-Hooks in 2005 used a tabu search heuristic to solve the TOP using an adaptive memory procedure to store and update solutions [31]. Archetti et al. in 2007 provided an improvement by developing two additional tabu search heuristics along with a variable neighborhood search [1].

3.3.4 Related Problems

In his study with Unmanned Surface Vessels (USV), Miller [24] references the TOPTW to help formulate what he calls the Unmanned Surface Vessel Observation Planning Problem (USVOPP). His problem involves optimized routing of USVs in two dimensions to maximize the values gained from target task completions. Specifically, the solution to the USVOPP provides multiple USV observation schedules to collect water temperatures to predict hurricane path and intensity as well as the occurrence of harmful algal blooms. In his work, he develops a mixed integer program to find the exact solution as well as a three phase algorithm to reach efficient solutions in terms of time. Negron [25] references Miller's work in to solve her Unmanned Aerial Vehicle (UAV) Planner Problem which looks maximizing values gained from target task completions using multiple heterogenous UAVs. She expands on Miller's work by planning in three dimensions, incorporating multiple locations for each task, and the development of a Composite Operations Planning Algorithm (COPA) using composite variables as inputs for a linear program.

Key differences with the CUPP posed in our thesis arise from allowing dual observations at targets to facilitate change detection or target refinement using different UAS payload capabilities, base swaps between modular Raven UASs, and plans for multiple sorties for each UAS. This thesis also advances current capabilities by incorporating a mix of both mandatory mission targets and optional LLO targets as well as the development of a string metaheuristic to solve large scale problems within operational time constraints. Units can seamlessly integrate our planner into their ISR Synchronization Process as described in section 2.3 to maximize use of UASs in a counterinsurgency. As with Negron's UAV Planner Problem, we also look at multiple heterogenous UASs in three-dimensional space.

3.4 Literature Review Conclusions and Approach

Based on our literature review, we can characterize the CUPP as a generalization of the Team Orienteering Problem with Time Windows (TOPTW). Whereas TOP does not allow a customer to be visited by more than one vehicle, our problem will allow dual observations of targets by different UASs as well as base swaps. Therefore, the most suitable problem definition for the CUPP combines opportunistic targets using TOPTW and mandatory primary targets using traditional routing problem methods. In addition, given increased complexity from our combinatorics problem, a metaheuristic such as the Iterated Local Search or Repeated Local Search shows promise to solve the problem efficiently.

[This Page Intentionally Left Blank]

4 Problem Formulation

This chapter introduces the assumptions and constraints of the Status Quo Concept and the Swap and Share Concept both graphically and with formulations for the exact approach. This chapter also describes the metaheuristics developed for both concepts in Matlab.

While the Status Quo Concept does not allow Raven UASs to collectively share primary mission targets and swap bases, the Swap and Share Concept removes these restrictions. Both concepts allow for the sharing of secondary LLO targets.

4.1 Status Quo Concept

This section provides an overview of the current status quo concept used by units according to current doctrine. As decentralized entities, Battalions and Companies who operate Raven UASs do not typically share mission targets with adjacent units. Units will only adopt other unit's mission targets during rare large-scale Brigade level operations. Therefore, even if one unit's mission target lies in close proximity in time and location to another unit's mission target, this Status Quo Concept ignores the potential advantages of shared target ownership to reduce total travel time. With this CONOP, UASs must return to their respective start base locations to complete a tour. This status quo concept will assume that units will include any available and feasible secondary LLO targets within the Brigade AO.

4.1.1 Graphical Representation

To illustrate the Status Quo concept, Figure 15 shows three different Raven units and their respective areas of operation. The Shadow unit encompasses the entire Brigade area of operations with its base and primary targets A, B, and C. Raven Unit 1 owns the Northwest sector with its base and primary targets R1A and R1B, Unit 2 owns the Northeast sector with its base and primary targets R2A and R2B, and Unit 3 owns the Southern sector with its base and primary targets R3A and R3B. For simplicity, we assume that this graph satisfies time windows and other target specific constraints. In the Status Quo concept, a unit will stay true to its primary targets but may deviate across boundaries in order to satisfy an LLO target. For example, Raven Unit 2 visits LLO target L2 in Raven Unit 1's area. Furthermore, if we suppose that primary targets R2B and R1B towards the northern edge of the graph were close in both proximity and time windows, the possibility of one Raven satisfying both targets would not be possible with this concept because of the strict ownership of primary targets.

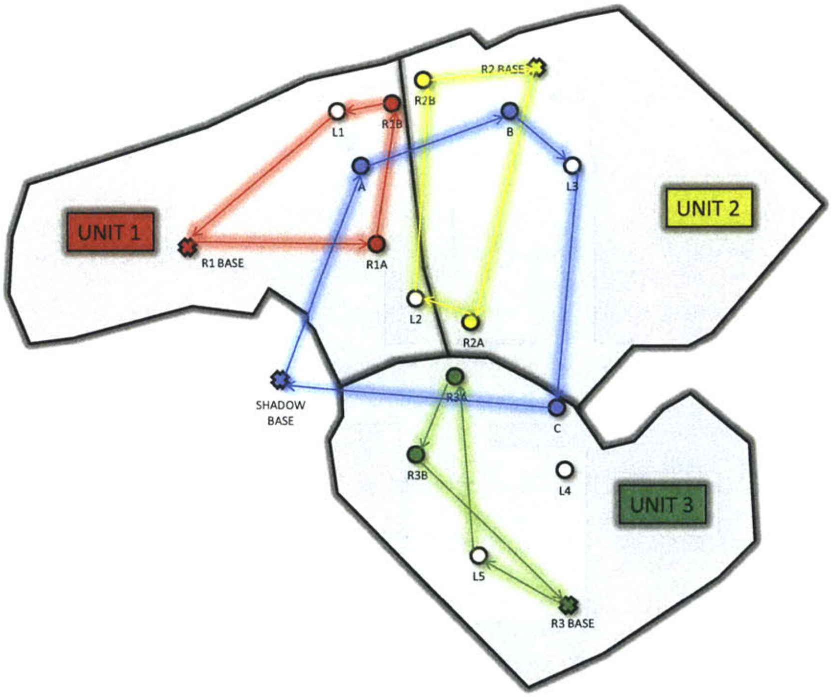


Figure 15 - Status Quo Graph Example. Raven UASs cannot collectively share primary targets and bases under this concept.

4.1.2 Model Formulation

As with the Vehicle Routing Problem, a Mixed Integer Programming (MIP) problem can be constructed to model our problem. Additionally, the exact solutions gained from our MIP will be useful for heuristic development and analysis.

4.1.2.1 Status Quo Problem Formulation

The construct of the Status Quo Concept follows:

Sets:

N: Set of all nodes
N_d: Set of all dual look nodes
N_s: Set of all single look nodes
T: Set of target locations
A: Set of all travel arcs
V: Set of all UASs

B: Set of all UAS bases
R: Set of all Raven UAS base locations, $R \subset B$
S: Set of Shadow base locations, $S \subset B$

Inputs:

$rvalue_i$: Reward value for visiting target i with Raven UAS.

$svalue_i$: Reward value for visiting target i with Shadow TUAS.

$dvalue_i$: Additional reward value for having two UASs observe dual look target

$endurance_v$: Endurance duration of UAS v

$traveltime_{i,j,v}$: Duration of time to travel from target i to target j for UAS v

$obstime_i$: Duration of time required to complete task at node i

$ETIOV_i$: Earliest time information of value for task at node i

$LTIOV_i$: Latest time information of value for task at node i

alt_i: Altitude required for task at node i (for noise considerations)

$speed_v$: Speed of UAS v

$maxalt_v$: Maximum altitude for UAS v

$minalt_v$: Minimum altitude for UAS v

$horizon$: Planning horizon

$earlystart_v$: Earliest time UAS v is available for missions based on maintenance requirements

$MaxWait$: Maximum wait time allowed at each node

Decision Variables:

$obs_{i,v}$: 1 if target i is observed by UAS v, 0 otherwise.

$travel_{i,j,v}$: 1 if arc i,j is traveled by UAS v, 0 otherwise.

$arrive_{i,v}$: A non-negative continuous variable that denotes the time UAS v arrives at target i

$depart_{i,v}$: A non-negative continuous variable that denotes the time UAS v departs target i

$wait_{i,v}$:	A non-negative continuous variable that denotes the time that UAS v waits at target i until start of observation
$dual_i$:	1 if dual look was accomplished at dual node, 0 otherwise

Maximize:

(1) Objective function

$$\max \sum_{i \in N} \sum_{j \in N} \sum_{r \in R} rvalue_i \, travel_{i,j,r} + \sum_{i \in N} \sum_{j \in N} \sum_{s \in S} svalue_i \, travel_{i,j,s} + \sum_{i \in N} dvalue_i dual_i = z^*$$

Constraints:

(2) Dual look Constraint

$$dual_i \leq .5 \cdot \sum_{v \in V} obs_{i,v} \quad \forall i \in N$$

(3) Maximum of two UASs can observe a Dual Node

$$\sum_{v \in V} obs_{i,v} \leq 2 \quad \forall i \in N_d$$

(4) Minimum Altitude constraint

$$alt_i \geq minalt_v \cdot obs_{i,v} \quad \forall i \in N, \forall v \in V$$

(5) Maximum Altitude constraint

$$alt_i * obs_{i,v} \leq maxalt_v \quad \forall i \in N, \forall v \in V$$

(6) Planning horizon constraint

$$arrive_{i,v} + obstime_i \leq horizon \quad \forall i \in N, \forall v \in V$$

(7) Endurance constraint

$$arrive_{i,v} + obstime_i \leq endurance_v \quad \forall i \in N, \forall v \in V$$

(8) Constraint to travel to each single look node with only one UAS

$$\sum_{v \in V} \sum_{i \in N} travel_{i,j,v} \leq 1 \quad \forall j \in N_s$$

(9) Balance Constraint for UASs

$$\sum_{i \in N} travel_{i,j,v} - \sum_{i \in N} travel_{j,i,v} = 0 \quad \forall j \in N, \forall v \in V$$

(10) Flow conservation constraint to ensure each Raven UAS leaves out of exactly one Raven base

$$\sum_{i \in R} \sum_{j \in N} travel_{i,j,v} \leq 1 \quad \forall v \in R$$

(11) Flow conservation constraint to ensure each Shadow UAS leaves out of exactly one Shadow base

$$\sum_{i \in S} \sum_{j \in N} travel_{i,j,v} \leq 1 \quad \forall v \in S$$

(12) Constraint to calculate time that UAS is ready to begin observation of node

$$arrive_{i,v} + wait_{i,v} \geq ETIOV_i \cdot obs_{i,v} \quad \forall v \in V, \forall i \in T$$

(13) UAS must depart after earliest allowable start time based on maintenance requirements

$$depart_{b,v} \geq earlystart_v \cdot travel_{b,j,v} \quad \forall v \in V, \forall j \in N, \forall b \in B$$

(14) Departure time constraint based on horizon

$$depart_{i,v} \leq horizon \cdot \left(\sum_{j \in N} travel_{i,j,v} \right) \quad \forall i \in N, \forall v \in V$$

(15) Departure occurs before the Latest Time Information of Value (LTIOV)

$depart_{i,v} \leq LTIOV_i \cdot obs_{i,v} \quad \forall v \in V, \forall i \in T$

(16) UAS cannot depart the node before sufficient time to observe the task has elapsed

$depart_{i,v} \geq arrive_{i,v} + wait_{i,v} + obstime_i \cdot obs_{i,v} \quad \forall i \in T, \forall v \in V$

(17) Constraint to ensure that there is sufficient travel time between tasks, where M denotes a sufficiently large number

$depart_{i,v} + traveltime_{i,j,v} - M(1 - travel_{i,j,v}) \leq arrive_{j,v} \quad \forall i,j \in N, \forall v \in V$

(18) Constraint to ensure Shadow TUAS cannot return to a Raven base

$$\sum_{v \in S} travel_{i,j,v} = 0 \quad \forall i \in N, \forall j \in R$$

(19) Constraint to ensure Raven UAS cannot go to and from Shadow base

$$\sum_{v \in R} travel_{i,j,v} = 0 \quad \forall i \in N, \forall j \in S$$

(20) Constraint to ensure UASs do not travel to same node consecutively

$travel_{i,i,v} = 0 \quad \forall i \in N, \forall v \in V$

(21) Constraint to ensure that UASs cannot return to another UASs base

$travel_{i,j,v} = 0 \quad \forall i \in N, \forall j \in B: j \neq v, \forall v \in V$

(22) Constraint to ensure wait time satisfies time window requirement

$depart_{i,v} + traveltime_{i,j,v} \cdot travel_{i,j,v} + wait_{j,v} \geq ETIOV_j \cdot travel_{i,j,v}$

$\forall i,j \in N, \forall v \in V$

(23) Constraint to ensure that wait time does not exceed maximum wait time

$wait_{i,v} \leq MaxWait \quad \forall i \in T, \forall v \in V$

(24) Binary constraint for decision variable

$obs_{i,v} \in \{0,1\} \quad \forall i \in N, \forall v \in V$

(25) Binary constraint for decision variable

$travel_{i,j,v} \in \{0,1\} \quad \forall (i,j) \in A, \forall v \in V$

(26) Non-negativity constraint for arrival time

$arrive_{i,v} \in \mathcal{R}^+ \quad \forall i \in N, \forall v \in V$

(27) Non-negativity constraint for departure time

$depart_{i,v} \in \mathcal{R}^+ \quad \forall i \in N, \forall v \in V$

(28) Non-negativity constraint for wait time

$wait_{i,v} \in \mathcal{R}^+ \quad \forall i \in N, \forall v \in V$

Minimizing the total duty time means minimizing the usage time for each UAS and associated support personnel. In order to do this, we solve a second MIP in order to complete our sequential multiple objective optimization:

$$Minimize \sum_{b \in B} \sum_{v \in V} arrive_{b,v} - \sum_{b \in B} \sum_{v \in V} depart_{b,v}$$

Subject to same constraints from (2) to (28) as above along with:

(29) Additional constraint to ensure objective value of new solution > z^*:

$$\sum_{r\in R}\sum_{i\in N}\sum_{j\in N} rvalue_i\, travel_{i,j,r} + \sum_{s\in S}\sum_{i\in N}\sum_{j\in N} svalue_i\, travel_{i,j,s} + \sum_{i\in N} dvalue_i dual_i \geq z^*$$

where z^* is the optimal objective function value of the first problem.

4.1.3 Heuristic Development for Status Quo Concept

Due to the complex combinatorial problem posed by the Status Quo Concept, we explore the development of heuristics to solve our problem when posed with a realistic problem size. To explore the solution space for the Status Quo concept, neighborhood solutions are found using the intra-route insertion and intra-route improvement heuristics.

4.1.3.1 Insertion Based Construction Heuristic

Developed in Matlab, our construction heuristic iteratively inserts available nodes into each UAS route while checking feasibility conditions prior to each insertion.

As a first step, all primary mission nodes for each UAS are sorted in ascending order by ETIOV and inserted in order based on feasibility. Each potential LLO to be inserted after this first step must satisfy the time constraints based on what is selected as the current 'recent' node and the 'next' node. For this heuristic, we assign the earliest available start time, $earlystart_v$, as the start time for each UAS.

In order to determine which LLO to select for insertion, a ratio is calculated for each potential visit. The LLO with the highest ratio value will be selected for insertion. Because we consider the time consumption of an insertion as less relevant than the actual value, the square of the value is applied in the ratio. The ratio of inserting a potential LLO node j between node i and node k along a Shadow TUAS's path is computed as:

$$(30)\quad Ratio_{ijkv} = \frac{(svalue_j)^2}{obstime_j + wait_j + LTIOV_j + traveltime_{ijv}}$$

where $obstime_j$ denotes the observation time, $wait_j$denotes the wait time, and $LTIOV_j$ denotes the Latest Time Information of Value at node j. The ratio reflects the concept that smaller observation times, wait times, earlier LTIOV, and smaller travel times are considered more attractive for insertions since it leaves availability for additional future insertions. The value is dependent on the type of node and type of UAS observing it. The same ratio is used for the Raven UAS using the $rvalue$.

At each potential insertion, feasibility is checked at each node in the route by calculating completion time, $ctime$, and wait time in order to check for endurance constraints and violation of time windows. Given that $x = 1$ if $ctime_i + traveltime_{ijv} \geq ETIOV_j$ and 0 otherwise:

$$(31)\quad ctime_j = \begin{cases} ctime_i + traveltime_{ijv} + obstime_j, & if\ x = 1 \\ ETIOV_j + obstime_j & , otherwise \end{cases}$$

$$(32)\quad wtime_j = \begin{cases} 0 & , if\ x = 1 \\ ctime_j - obstime_j - ctime_i - traveltime_{ijv}, & otherwise \end{cases}$$

If given two primary targets, a potential LLO may have a time window spanning almost the entire time horizon. In Figure 16, the LLO observation time window might be only five minutes but the time window spans from 0 to 75 minutes. Insertion of the LLO will be feasible only if the travel times, wait times, and observation times all meet the constraints given by the problem.

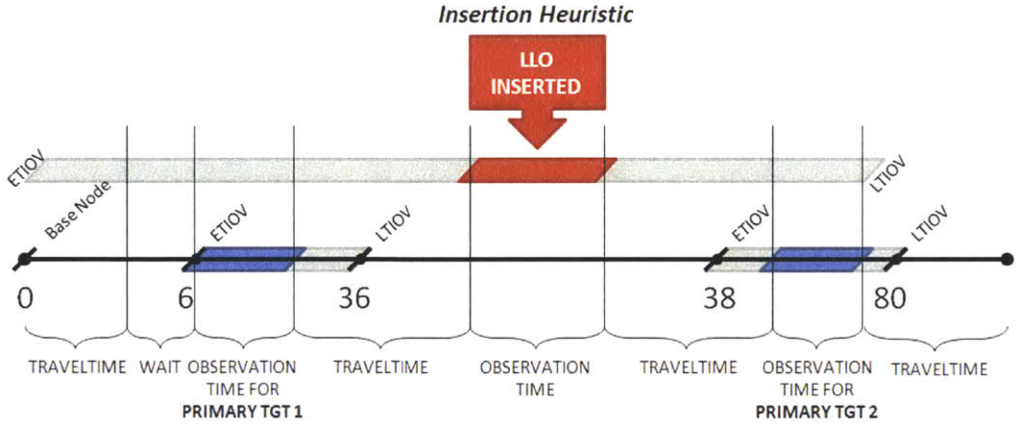


Figure 16 - Insertion Example. Although the time window for the LLO extends across nearly the entire horizon, the only feasible time is in between the two primary targets.

Once an LLO is inserted, this LLO becomes the new 'next' node and we repeat the process until no other available LLOs are found. Once a single look LLO is visited, it is no longer available. If a dual LLO is observed, a counter within the heuristic tracks the visits in order to ensure each dual LLO is observed at most twice.

The Insertion Based Construction Algorithm can be summarized with the following pseudocode:

```
for each UAS $v$
Sort Primary targets for UAS $v$ in ascending order based on ETIOV
    and insert if feasible
Initialize $v = 1$, $position\ of\ route\ node\ k = 1$, $InsertAt = k + 1$
        while $k <$ number of node stops in Route
         $num\_inserts = 0, recent = kth\ position\ in\ Route$
                for all available LLOs
                        if feasible and available LLO targets exist when positioned
                        at $InsertAt$, compute LLO target ratios:
                        $Ratio_{ijkv} = \frac{(value_j)^2}{obstime_j + wtime_j + LTIOV_j + traveltime_{ijv}}$
                        end if
                end for
                while positive ratio LLO targets exist,
                Add max ratio LLO $l$ to UAS $v$ target deck
```

```
                if 1st visit to dual look target, set dual counter to 1
                elseif part of dual look and looked at before, remove from
                avail list
                else 1st visit and single look target, remove from avail list
                end if
                    for all available LLOs,
                        if feasible and available LLO targets
                    exist when positioned at $InsertAt$, compute
                    LLO target ratios:
                        $Ratio_{ijkv} = \frac{(value_j)^2}{obstime_j + wtime_j + LTIOV_j + traveltime_{ijv}}$
                    end if
                    end for
            end while
    Update $InsertAt = InsertAt + num\_inserts + 1, k = k + num\_inserts + 1$
    end while
    Get BestRoute for UAS $v$
end for
```

The flow chart in Appendix A illustrates the Insertion Based Construction Heuristic Algorithm in detail. This algorithm will be used to construct the initial UAS routes for both the Status Quo and Swap and Share Concepts.

4.1.3.2 2 Opt Intra-route Improvement Heuristic

In order to improve the route constructed by the Insertion Based Construction Heuristic, we apply an intra-route swap heuristic that reduces the cost of the route by swapping the positions of a pair of nodes in the route. For our problem, the cost of the route translates to the route's travel time. Due to the limitations presented by the time windows, relocating extreme nodes in terms of route positions will most likely prove infeasible, especially for routes that contain multiple node stops. Therefore we only look at executing the procedure for the nearest n neighbors to the candidate node. For example, if we choose parameter n to be .25, and the length of the route is 10 nodes, we take the length of the route and multiply it by n to get 2.5 or after rounding up to the nearest integer, 3 nearest neighbors. The heuristic checks for feasibility with each swap. If feasible, the new route is chosen as the best route if it provides a shorter travel time.

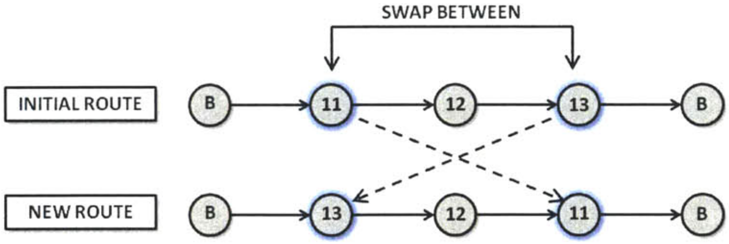


Figure 17 – Example of Intra-route 2 Opt Improvement. Here 'B' represents the base nodes.

The flow chart in Appendix B illustrates the Intra-route 2 Opt Improvement Heuristic Algorithm in detail.

In summary, the Status Quo Metaheuristic executes the following steps:

1. Construct initial route using Insertion Based Construction Heuristic
2. Improve with Intra-Route 2 Opt Heuristic
3. Reapply Insertion Heuristic with improved route

The Status Quo Metaheuristic steps are illustrated in Figure 18:

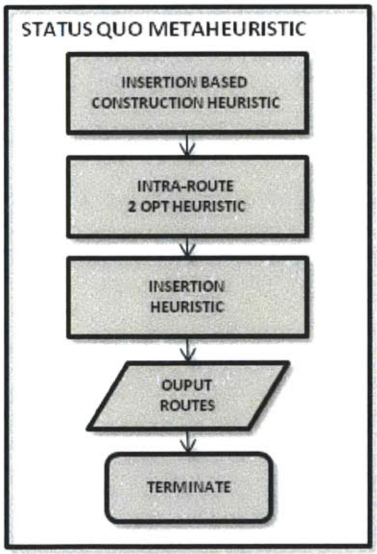


Figure 18 – Status Quo Metaheuristic Flow. Each step accepts the previous step's route solution as its input.

4.2 Swap and Share Concept

In contrast to the Status Quo CONOP, the Swap and Share CONOP allows the sharing of targets between units. Since the Shadow TUAS already operates as a Brigade asset across units, this Swap and Share CONOP focuses primarily on the routes of Raven UASs. With identical Raven systems, sharing of targets will not result in a loss of target coverage quality as long as units provide detailed guidance for their respective targets. In addition, because the modularity of Raven UASs supports minimal maintenance and turnaround time, this CONOP will also explore the advantages of swapping bases, provided that units maintain a one for one exchange at the end of the mission cycle. Therefore unlike the Status Quo CONOP, this will allow for Raven UASs to form paths in addition to tours.

4.2.1 Graphical Representation

To illustrate the Swap and Share concept, we revisit the example graph from the Status Quo concept. With this concept, Raven UASs cooperatively visit any of the other Ravens' primary targets as well as base nodes. For example, in Figure 19, Raven 2 leaves from its base in the Northeast sector, visits its primary target R2B, enters unit 1's sector, visits R1B, L1, and completes its sortie at Raven 1's base location. Upon completion of the missions, each unit maintains the same number of Raven UASs.

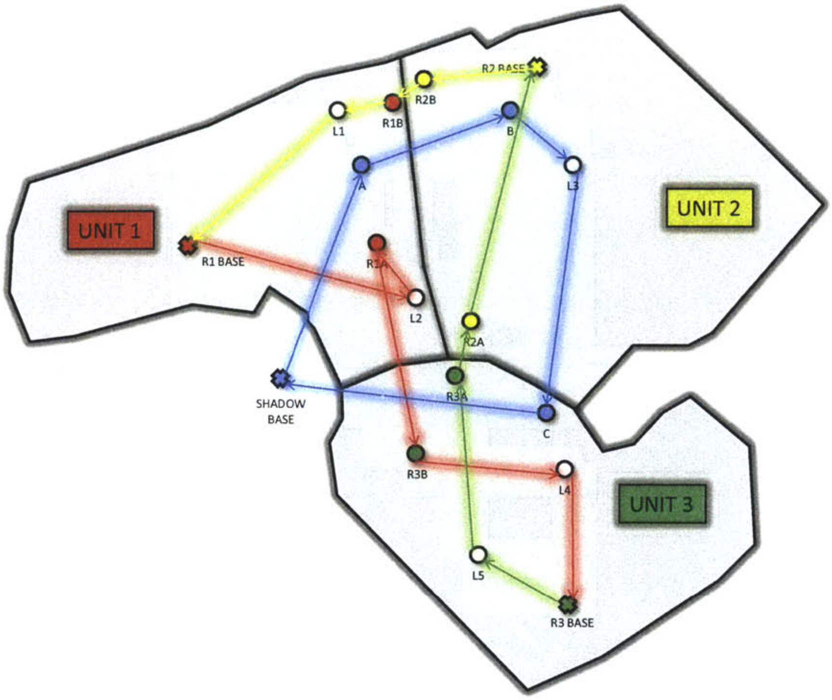


Figure 19 - Swap and Share Graph Example. Raven UASs share both primary targets and bases under this concept allowing linear paths to be feasible in addition to tours.

4.2.2 Model Formulation

Because the Raven UASs are not constrained to their respective original base locations, this formulation must include an additional flow conservation constraint for each base. We make the assumption that the number of Raven UASs at each Raven unit experiences no net change at the end of the planning horizon. Otherwise a unit could potentially result in losing all of their Raven UASs to adjacent units after a few mission sequences.

4.2.2.1 Swap and Share Problem Formulation

For this model formulation, we adjust the status quo formulation with the following:

To allow any Raven UAS to return to a Raven UAS base, we include the following constraint:

(33) Flow conservation constraints to ensure that only one Raven UAS returns to each Raven base

$$\sum_{i \in N} \sum_{j \in R} travel_{i,j,v} \leq 1 \quad \forall v \in R$$

Since paths can be formed with this concept in addition to tours, we substitute the balance constraint (10) with the following equations:

(34) Force an arc to leave every observed target node

$$\sum_{j \in N} travel_{i,j,v} - obs_{i,v} = 0 \quad \forall v \in V, \forall i \in T$$

(35) Force an arc to enter every observed target node

$$\sum_{j \in N} travel_{j,i,v} - obs_{i,v} = 0 \quad \forall v \in V, \forall i \in T$$

We also remove the base restriction constraint to allow swapping of bases.

(21) Constraint to ensure that UASs cannot return to another UASs base

$travel_{i,j,v} = 0 \quad \forall i \in N, \forall j \in B: j \neq v, \forall v \in V$

4.2.3 Heuristic Development for Swap and Share Concept

The Swap and Share Concept Heuristic adds three more improvement heuristics to the heuristics used for the Status Quo concept: Deletion Insertion Inter-Route Improvement, 2-Exchange Inter-Route Improvement, and a UAS Base Inter-Route Improvement. This concept executes the original construction heuristic, intra-route 2 opt insertion heuristic, and insertion heuristics exactly as before.

4.2.3.1 Deletion Insertion Inter-Route

The Deletion Insertion inter-route heuristic takes the improved UAS routes from the 2-Opt heuristic and for each possible pair of routes, deletes a node from one node and inserts it in another. Just as with the intra-route swap heuristic, we only look at executing the procedure for the nearest neighbors within parameter *n* to the candidate node due to time window limitations. The heuristic checks for feasibility after each relocation and accepts the change if it results in a shorter total travel time. A special case of the Deletion-Insertion heuristic occurs when a route containing only one target node is selected for deletion. In this case, the route essentially disappears. For our problem, this can only occur with the Raven UAS because only the Ravens can share primary targets according to our concept. Additionally, if a base node is selected for deletion and is inserted next to the adjacent route's base node, the heuristic reverts back to its original routes.

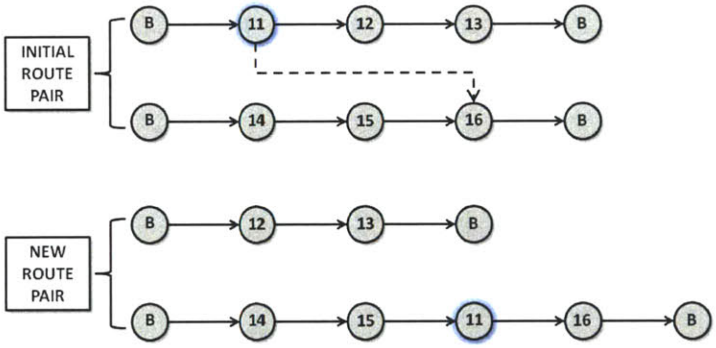


Figure 20 - Example of Deletion Insertion Inter-Route Improvement. Here Node 11 is deleted from the first route and inserted into the second route before node 16 based on feasibility with time windows.

The flow chart in Appendix C illustrates the Deletion Insertion Inter-Route Improvement Algorithm in detail as executed in Matlab. The initial route used as an input for this algorithm will be the routes obtained from execution of the 2 opt Intra-Route Improvement Heuristic.

4.2.3.2 2 Exchange Inter-Route

The 2 Exchange Inter-Route Heuristic takes two nodes from two different routes and exchanges their positions to reduce the total travel time of both routes. This heuristic executes every possible combination and checks for feasibility and reduced combined travel times as acceptance criteria for a new best route.

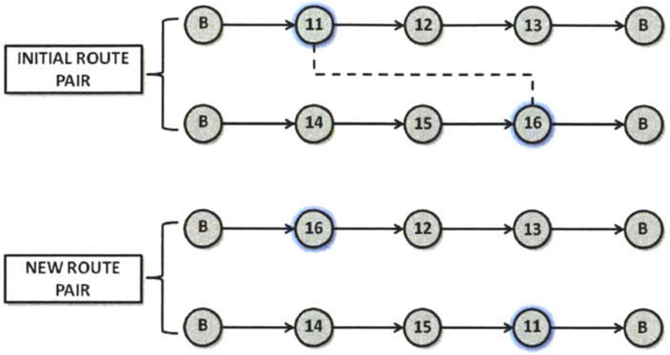


Figure 21 - Example of 2 Exchange Inter-Route Improvement. Here nodes 11 and 16 are chosen for an exchange and checked for feasibility and combined travel time reduction.

The flow chart in Appendix D illustrates the 2 Exchange Inter-Route Improvement Algorithm in detail. The initial route used as an input for this algorithm will be the routes obtained from execution of the Deletion Insertion Inter-Route Improvement Heuristic.

4.2.3.3 UAS Base Swap Heuristic

In order to look at all feasible Raven UAS base swaps, our UAS Base Swap Heuristic finds all of the base options for the start of the route and end of the route that are closer in travel time than the original. A verification list created by the permutation of the possible base allocations is used to check for feasibility. For example, for a simple two Raven UAS scenario, the verification list would consist of $n!/(n-k)!$ subsets or (2,3), (3,2) for Raven UAS's whose original base nodes were {2,3}.

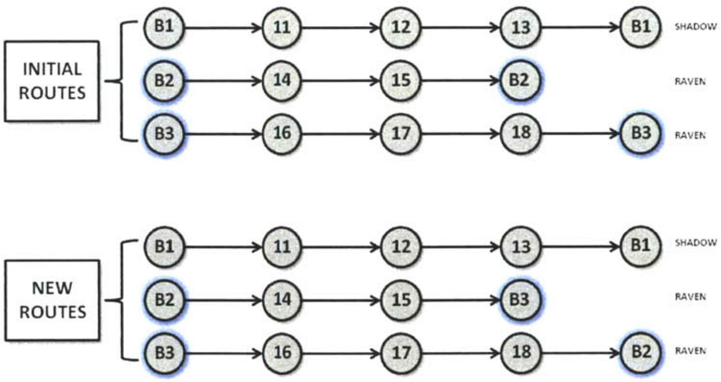


Figure 22 - Example of UAS Base Swap Improvement. Here, the two Raven UASs in base 2 and 3 swap bases at the conclusion of their respective missions. The Shadow UAS in base 1 cannot swap with the Raven base according to our assumptions. Each base does not experience a net change in the number of Ravens.

4.2.3.4 Total Duty Time Minimization

In order to minimize the total duty time and minimize the usage time for each UAS and associated support personnel, we incorporate a step within the algorithm to maximize the start time of each UAS while maintaining feasibility of the route. The maximum start time for each UAS, which we annotate as $ctime_1$ denoting the completion time for the first base node in the route, is:

(36) $ctime_1 = LTIOV_2 - obstime_2 - traveltime_{1,2,v}$

where $LTIOV_2$ and $obstime_2$ denote the Latest Time Information of Value and observation time for the second node in the route, and $traveltime_{1,2,v}$ denotes the travel time from the first node to the second node in the route with UAS v. Because the earlier heuristics already solved for a feasible solution using the input $earlystart_v$ for the earliest available start time for UAS v, this step iteratively reduces the maximum start time determined from equation 36 down to at most $earlystart_v$ in order to find the best start time that minimizes overall duty time. With each reduction, the completion time, wait time, and feasibility are checked for the route.

4.2.3.5 Repeated Local Search

We pursue diversification by introducing a level of randomness with a Repeated Local Search algorithm, a relatively straight forward iterative hill climbing technique to maximize the objective function. With this algorithm, instead of choosing to insert the LLO target with the maximum ratio described with equation 30, we allow a suboptimal LLO target to be inserted given a certain parameter p defined as the probability of acceptance as a number between 0 and 1. We only accept a move from the improvement heuristics with parameter p. For example, for our insertion heuristic, there may be multiple potential LLOs for insertion and associated ratios. Our heuristic only accepts the LLO associated with the maximum ratio whenever a pseudorandom number uniformly generated between 0 and 1 is less than p. If this number is greater than p, we pick one of the other LLOs with a lower ratio value. The termination criteria used for this metaheuristic will be the maximum iteration limit denoted as $MaxIter$ as well as the number of iterations with no improvement denoted as $NumIterNoImp$, whichever comes first.

The overall Repeated Local Search algorithm can be described as follows:

```
Repeated Local Search
BestFoundVal = Objval;
BestFoundRoute = BestRoute;
MaxIter = 50;
NumIterNoImp = 0;
while NumIterNoImp < MaxIter do
    Execute Swap and Share Metaheuristic
    Minimize wait time
    if Solution better than BestFoundVal then
        BestFoundVal = Solution;
        BestFoundRoute = Solution's route;
        NumIterNoImp = 0;
    else
        NumIterNoImp +1;
    end
end
Return BestFoundVal and BestFoundRoute
```

The overall Swap and Share metaheuristic is shown in Figure 23:

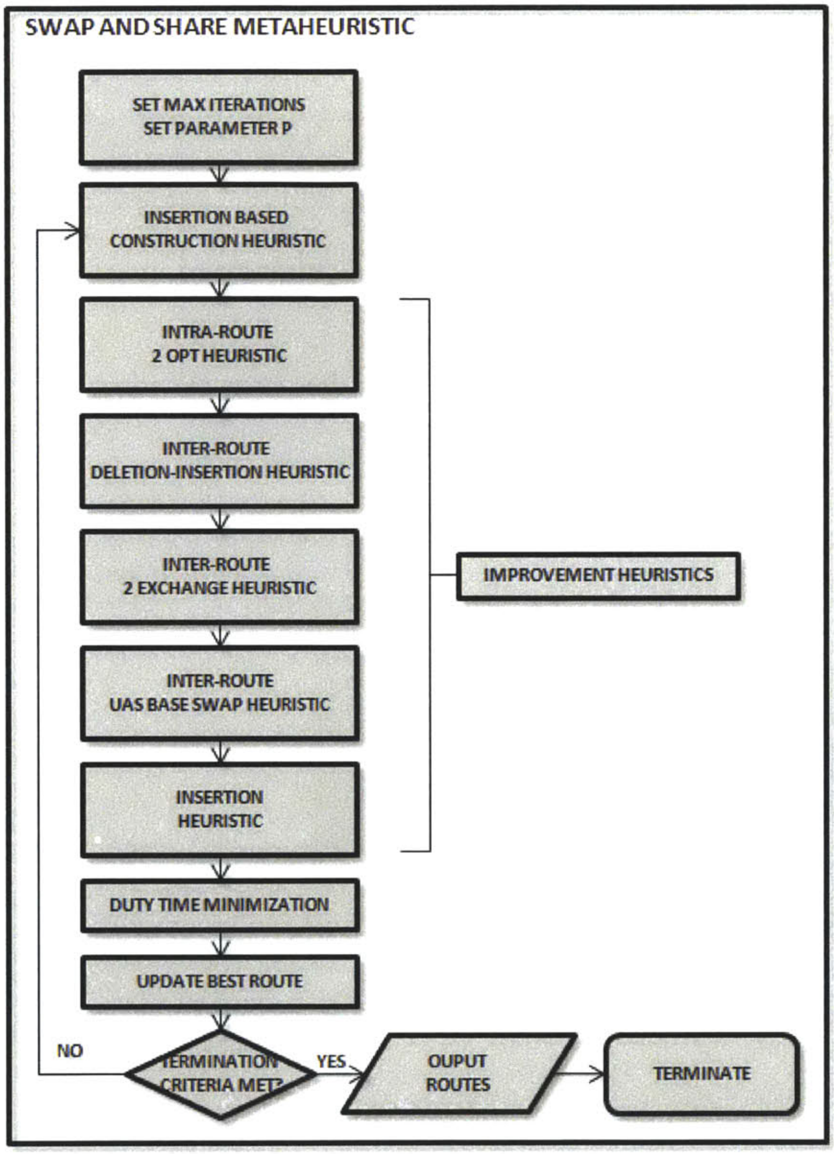


Figure 23 – Swap and Share Metaheuristic.

Table 3 provides a one look summary of each of the heuristics used along with the concept application.

Type	Heuristics	Description	Status Quo	Swap and Share
Insertion	Insertion Based Construction	Constructs initial routes with UAS primary targets and inserts LLO targets	X	X
	Insertion	Inserts LLO targets into best route	X	X
Intra-Rte	2 opt	Swaps two nodes from single UAS route	X	X
Inter-Rte	Deletion Insertion	Deletes a node from one route to insert into another route		X
	2 exchange	Swaps two nodes from distinct UAS routes		X
	Base swap	Swaps base stop nodes with all UAS routes, checking all permutations		X

Table 3 - Heuristic Summary with Concept Application. While both concepts use the same insertion and intra-route improvement heuristics, the randomness introduced by the Swap and Share Concept's Repeated Local Search algorithm allow for consideration of different candidate nodes.

5 Testing and Analysis

This chapter focuses on the advantages gained by the Swap and Share Concept as well as an evaluation of the metaheuristics developed. Due to the unavailability of real time data inputs, we conduct multiple experiments with randomized data as well as pseudo-realistic data based on a map overlay of Baghdad, Iraq and likely target locations and bases from terrain analysis.

5.1 Testing Methodology

We carried out the computational experiments using OPL Studio and Matlab on a PC with a 2.8 GHz Intel Core 2 Duo processor and 4GB of RAM. IBM's ILOG OPL Studio version 12.2 uses an optimization solver called CPLEX to solve the MIP by initially reducing the problem size through various preprocessing steps. After finding the optimal objective value of the linear programming relaxation using the simplex method, CPLEX then chooses the best integer solution using a branch and bound algorithm. With large size problems, the RAM limitations can cause CPLEX to terminate before reaching the optimal solution. We used Matlab version 7.12 to run the metaheuristics for both concepts.

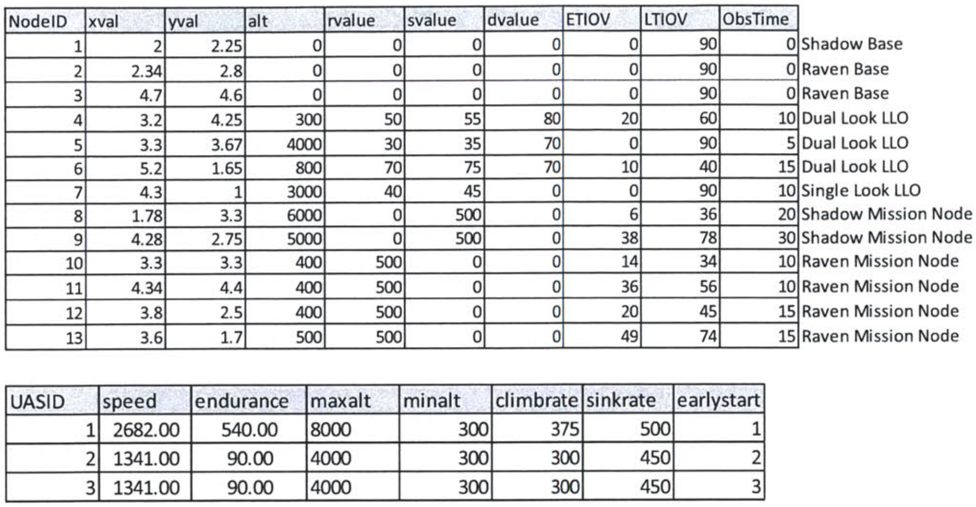
5.1.1 Pseudorealistic Data

In order to replicate realistic data, we used imagery from Google Earth depicting the city of Baghdad, Iraq and a typical Brigade size Area of Operations (AO) approximately 25 km by 18 km with Raven targets situated in the vicinity of Company and Battalion AO's and Shadow targets situated in and around the larger Brigade AO. We scattered LLO targets across the Brigade AO in likely LLO target locations based on terrain analysis for the possible types of targets described in section 2.4.1. Each grid length is approximately 3.6 km. Units are converted to use meters for distance and minutes for time. Reflecting realistic operations, we classified approximately 40% of the nodes as primary target nodes and 60% of the nodes as LLO secondary target nodes. Primary mission nodes were given values of 500 to distinguish them from the LLO targets with values ranging from 5 to 95, ostensibly based on the Effects Working Group's discussion. Initially we set the maximum number of dual look targets to three for our data sets. The baseline reference map is shown in Figure 24:

Figure 24 - Notional Baseline Baghdad map used for test sets. Each of the three sectors represents a Battalion area of operations. The entire area represents the Brigade area of operations.

5.1.2 Inputs and Parameters

For the input data format, we use excel to provide the UAS and node attributes. We keep the UAS parameters constant for our experiments, using an average speed of 100 mph for the Shadow TUAS, equivalent to 2682 m/min and 50 mph for the Raven UAS, equivalent to 1341 m/min. Maximum altitude is set according to operational altitude air corridors and the other attributes follow the reference data given in Figure 5. For our test sets, we use arbitrary earliest available times, setting UAS v's early start time as v. An example input from Excel is shown in Table 4 with labels:



NodeID	xval	yval	alt	rvalue	svalue	dvalue	ETIOV	LTIOV	ObsTime	
1	2	2.25	0	0	0	0	0	90	0	Shadow Base
2	2.34	2.8	0	0	0	0	0	90	0	Raven Base
3	4.7	4.6	0	0	0	0	0	90	0	Raven Base
4	3.2	4.25	300	50	55	80	20	60	10	Dual Look LLO
5	3.3	3.67	4000	30	35	70	0	90	5	Dual Look LLO
6	5.2	1.65	800	70	75	70	10	40	15	Dual Look LLO
7	4.3	1	3000	40	45	0	0	90	10	Single Look LLO
8	1.78	3.3	6000	0	500	0	6	36	20	Shadow Mission Node
9	4.28	2.75	5000	0	500	0	38	78	30	Shadow Mission Node
10	3.3	3.3	400	500	0	0	14	34	10	Raven Mission Node
11	4.34	4.4	400	500	0	0	36	56	10	Raven Mission Node
12	3.8	2.5	400	500	0	0	20	45	15	Raven Mission Node
13	3.6	1.7	500	500	0	0	49	74	15	Raven Mission Node

UASID	speed	endurance	maxalt	minalt	climbrate	sinkrate	earlystart
1	2682.00	540.00	8000	300	375	500	1
2	1341.00	90.00	4000	300	300	450	2
3	1341.00	90.00	4000	300	300	450	3

Table 4 - Example Input Tables from Excel. Realistically, each node's value would come from the Effects Working Group as described in Section 3.1.3. The rvalue, svalue, and dvalue represent the values gained by observing the node for a Raven UAS, Shadow UAS, and as dual look, respectively.

For both the 2-opt and Deletion-Insertion heuristics that use the parameter we originally defined in section 4.1.3.2 as n to determine the neighboring nodes to consider, we will vary n from 0 to 1 in increments of .1 to determine the lowest value that results in the same objective value.

5.1.3 Measures of Performance

As measures of performance, we will use the following:

1. Objective function value: To increase the overall situational awareness for the commander in COIN by observing as many primary and LLO targets as possible.
2. Computation time (seconds): To determine if the time required to reach the optimal objective function value meets operational constraints.
3. Total duty mission time for all of the UASs (minutes): To minimize the cost in terms of both personnel and equipment for all of the UASs.

4. Total wait time for all of the UASs (minutes): To minimize the idle wait time at each node in order to maximize UAS capability during each mission.

5.1.4 Hypotheses

Our hypotheses focus on the advantages of swapping bases and collective sharing of Raven UAS targets. We also analyze how the MIP and metaheuristics perform as we modify the types and numbers of targets. We hypothesize the following:

1. The Swap and Share Concept will provide an overall improved objective function value over the Status Quo concept by allowing for shorter travel times and observation of additional LLO targets.
2. The average computation time to reach the exact solution using the MIP will increase as we increase the number of primary and LLO targets. This will apply to both the Swap and Share Concept and Status Quo concept outlined in section 4.1.2.1 and 4.2.2.1, respectively.
3. Increasing the number of dual look targets will significantly increase run times for the exact approach with a lesser impact on the metaheuristic run times.
4. The Swap and Share Concept's metaheuristic will reach the optimal objective function value in the majority of cases while dramatically reducing the computation time.
5. Allowing the collaborative sharing of targets for the Raven UASs while restricting the swapping of bases will achieve the same objective function value as allowing collaborative sharing and base swapping for majority of cases.

6. The Swap and Share metaheuristic will achieve objective values within a 5% optimality gap for random target sets scattered both uniformly and in a more linear border scenario.

5.2 Results and Analysis

5.2.1 Experiment 1

For our first experiment, we determine the largest problem size the MIP can handle, along with the improvement gained with the Swap and Share Concept. We first tested the Status Quo and the Swap and Share Concepts MIP formulations with 20 different inputs, varying the number of UASs from 2 to 5 and the number of targets from 5 to 25 in increments of 5. We used pseudorealistic data based off of terrain analysis for the locations of bases and targets. In addition to getting the exact solutions to use as a basis for analyzing our metaheuristics, this allowed us to test our first hypothesis regarding the advantages gained from allowing a collective sharing of targets and swapping of bases. It also allowed us to test our second hypothesis to see if average run times increase as a function of the number of targets. We present our results in Figure 25.

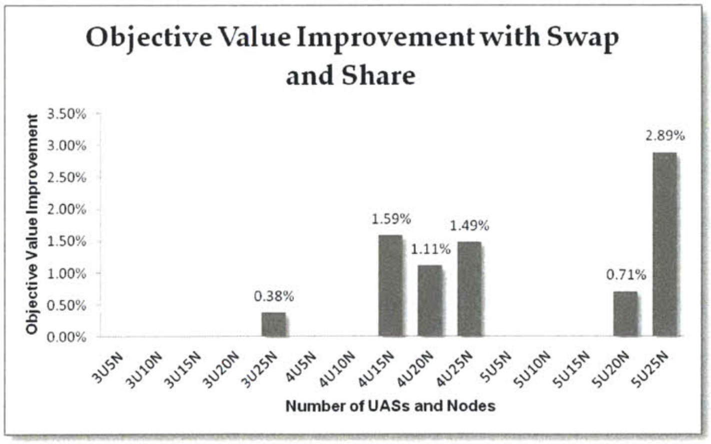


Figure 25 - Objective Value Improvement with Swap and Share. In general, as the number of UASs and targets increase, the advantages of the collective sharing and swapping of bases increase. For the 5U25N (5 UAS, 25 Target) case, there was a 2.89% increase in the Objective Value or a 190 point increase compared to the Status Quo Concept results.

From the objective value improvement in Figure 25, we can substantiate our first hypothesis; the Swap and Share Concept provides a higher objective value especially given a bigger problem size. As expected, the objective values for the 2 UAS tests are the same for both concepts because no swapping between the Shadow TUAS and Raven UASs can occur. Sharing targets and bases provide more options for each UAS, allowing for shorter travel times and the inclusion of additional LLO targets. Figure 25 shows an objective function improvement compared to the Status Quo Concept results in 6 out of 20 cases with no change in the other 14 cases. Each improvement in value results from the addition of one to three LLO nodes. The Swap and Share concept resulted in an overall objective value improvement of 460 with the most dramatic improvement involving the biggest problem size at 5 UASs and 25 target nodes. Operationally, given that Brigades own a Shadow TUAS system and 15 Raven systems, most Brigades operate an average of four or more UASs during any given horizon, or approximately one for each Battalion. The results show that in half of the cases with 4 or more UASs, the Swap and Share concept results in improved coverage of targets with no change in the other half. In two of these cases, Raven UASs swapped bases to achieve higher objective function values. By minimizing total travel time between targets, this concept creates opportunities to observe additional secondary LLO targets.

For example, with the 4U20N (4 UAS, 20 target nodes) case, the Status Quo Concept visits node 7, whereas the Swap and Share Concept visits nodes 9 and 12 in lieu of node 7 resulting in a difference in value of 60 or a 1.11% improvement. We show the differences in Figures 26 and 27:

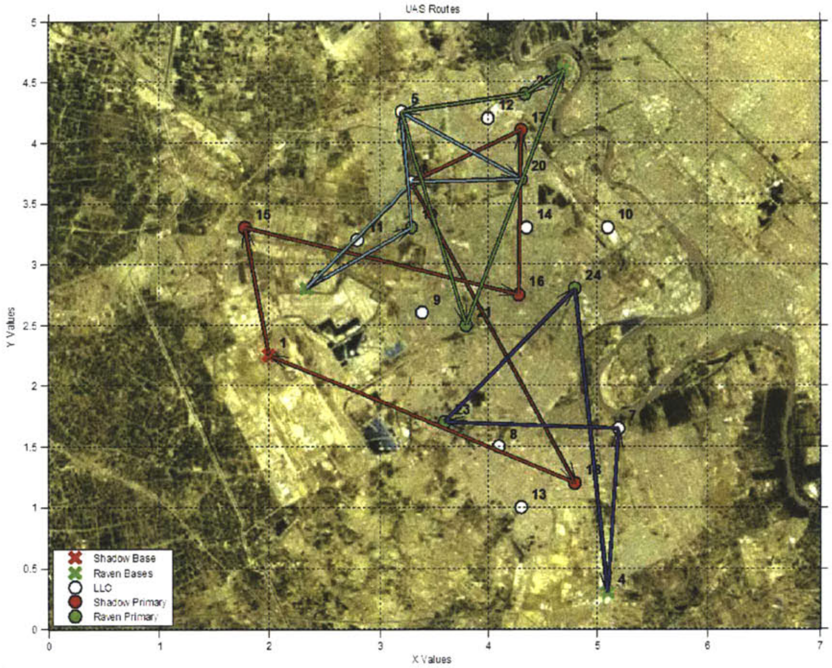


Figure 26 - Status Quo Concept's solution to the 4U20N case. Note that each UAS must return to its respective base.

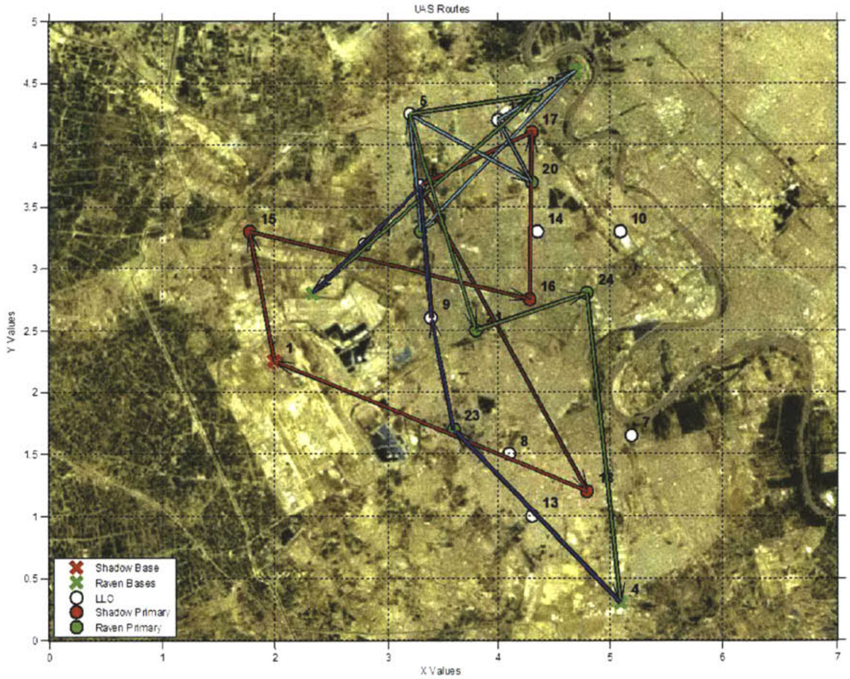


Figure 27 - Swap and Share Concept's solution for the 4U20N case. For the same example case used in the Status Quo Concept, the Swap and Share concept minimizes travel time by allowing the swapping of bases resulting in an objective value improvement of 60.

From Figure 28, we can also substantiate our second hypothesis that average run times generally follow an increasing function as the number of targets per UAS increases. As a penalty for searching through more of the solution space, the exact solution to the Swap and Share concept requires an increasing amount of time given more options to consider. We can also conclude that average run times appear somewhat insensitive to increasing numbers of UASs for a constant number of targets. Figure 29 in fact, shows that a reduction in run time can occur (as in the transition from 15 nodes with 4 UAS to 5 UAS) as more UASs become available to reconcile the same number of targets. Errors for insufficient memory occurred with the ILOG OPL software when trying to solve larger problems with 4 and 5 UASs with 20 or more targets, taking over half a day to compute.

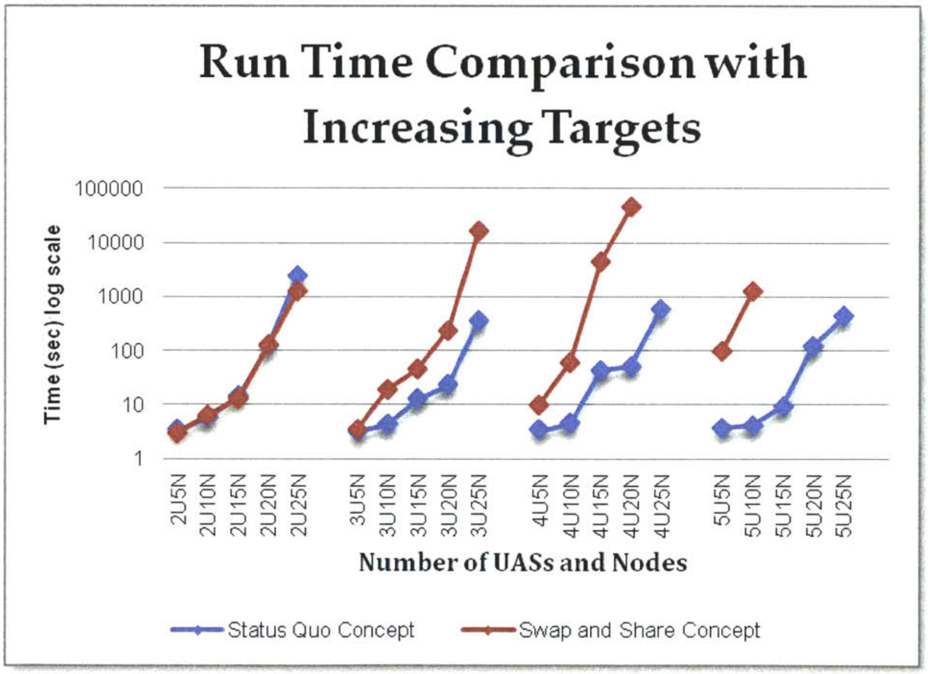


Figure 28 - Run Time Comparison of the exact approach between the Status Quo Concept and Swap and Share Concept as we increase the number of targets for each UAS set. The time required to reach the exact solution increases as a function of the number of targets. The missing data points reflect the problem sizes that CPLEX could not solve due to insufficient memory.

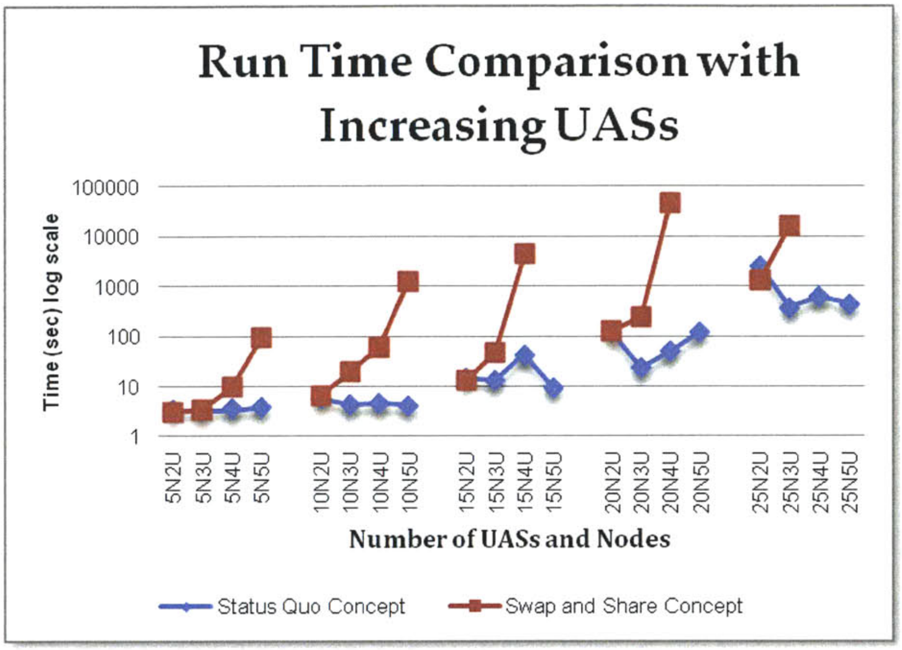


Figure 29 - Run Time Comparison when increasing the number of UASs and keeping the number of targets constant. Note that with the Status Quo Concept, increasing the number of UASs can lead to a decrease in time as more options become available.

5.2.2 Experiment 2

Next we address our third hypothesis and test how the number of dual look nodes impacts the run times for the exact solution and metaheuristic solution to the Swap and Share Concept. We focus our attention on a pseudorealistic 4 UAS 12 target node case because of its relevance to steady state operations and because of its tractability. Again, we restrict the Shadow TUAS from observing any additional LLO targets by constraining it to just one primary target that spans the length of the planning horizon. We present the findings in Figure 30. Here, our experiment counters our hypothesis that run times will increase as a function of the number of dual look targets. When the number of dual look targets shifts from 4 to 5, we note a decrease in run time due to the unequivocal advantage the new dual look target provides for an improvement of 80 in the Objective Value.

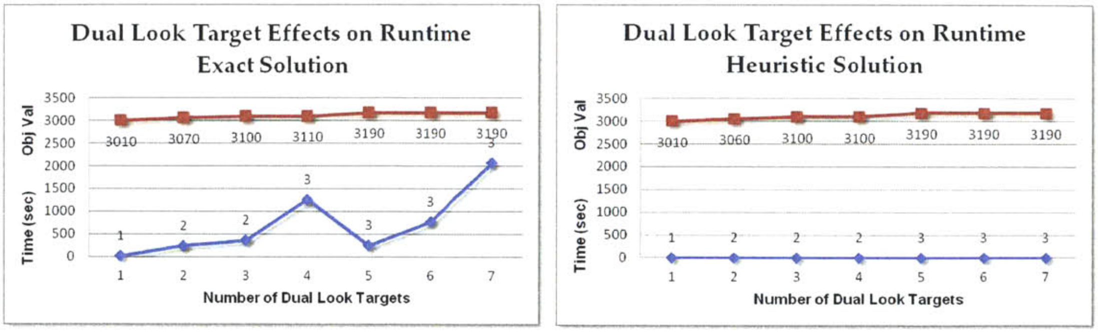


Figure 30 – Dual Look Target Effects on Run Time Performance Exact vs Metaheuristic Solutions. Using a 4 UAS 12 target node test case, the number of dual look targets satisfied for each test is annotated above each observation on the lower line plot. The reason for the run time decrease from 4 to 5 dual look targets is due to a clear advantage in objective function value provided by the fifth dual look node. The metaheuristic performs well, deviating from the optimal solution in 2 out of 7 cases. The largest run time difference was 34.3 minutes with an optimality gap of .33%

5.2.3 Experiment 3

As the next experiment, we compared our Status Quo metaheuristic and Swap and Share metaheuristic with their respective MIP exact solutions to test our fourth hypothesis which predicted a dramatic reduction in computation time. The time savings using the metaheuristics for both concepts proves significant compared to their respective MIP solution times, especially as the number of targets exceeds 20. For the 2 opt and Deletion-Insertion heuristics, we find that the value of our parameter $n > .1$ achieves the maximum objective function value for the different problem sizes.

To assess the performance of our metaheuristics, we define the optimality gap as:

$$(37) \quad Optimality\ gap = \frac{exact\ solution - metaheuristic\ solution}{exact\ solution}$$

We provide the results in the following tables and figures.

Tests	Avg Time (s)	CPU Time Diff	Mission Time (min)	Mission Time Diff	Wait Time (min)	Wait Time Diff	Obj Val	Optimality Gap
2U5N	0.12	-3.35	143.9	1.99	1.51	0	2225	0.00%
2U10N	0.36	-5.46	164.72	1	0.00	0	2685	0.00%
2U15N	0.67	-13.656	157.27	-4.15	9.08	9.08	4055	0.73%
2U20N	0.91	-118.67	169.82	2.88	23.03	21.28	5135	0.00%
2U25N	1.85	-2472.96	168.96	1.99	14.65	13.77	5210	0.19%
3U5N	0.23	-3.01	178.98	3	1.51	0	2220	0.00%
3U10N	0.62	-3.69	237.45	3	0	0	3210	0.00%
3U15N	1.21	-11.51	222.53	3	2.83	2.83	4315	0.00%
3U20N	1.42	-22.01	243.17	3	5.16	5.16	5215	0.00%
3U25N	2.42	-368.95	243.97	3.8	5.23	5.23	5145	1.34%
4U5N	0.22	-3.39	198.54	11.49	0	0	2225	0.00%
4U10N	0.68	-3.84	295.32	16.86	0	0	3355	0.00%
4U15N	1.36	-41.68	280.03	-24.07	0	0	4355	0.91%
4U20N	1.37	-49.26	294.82	4	2.83	2.83	5385	0.00%
4U25N	3.21	-600.95	294.82	4	2.83	2.83	5385	0.00%
5U5N	0.16	-3.58	174.33	5	0	0	2500	0.00%
5U10N	0.40	-3.78	286.53	5	0	0	3430	0.00%
5U15N	0.58	-8.74	355.27	23.48	0	0	4575	0.00%
5U20N	1.82	-119.94	408.37	22.53	6.09	6.09	5550	1.77%
5U25N	1.71	-440.64	415.25	16.88	14.24	9.51	6510	1.06%

Table 5 - Status Quo metaheuristic Results and Comparison to Status Quo MIP. Mission Times did not deviate more than 6% from optimal mission time. The difference columns show the metaheuristic times minus the MIP times. Wait times increased for 10 out of the 20 test cases, mostly involving 15 or more target nodes.

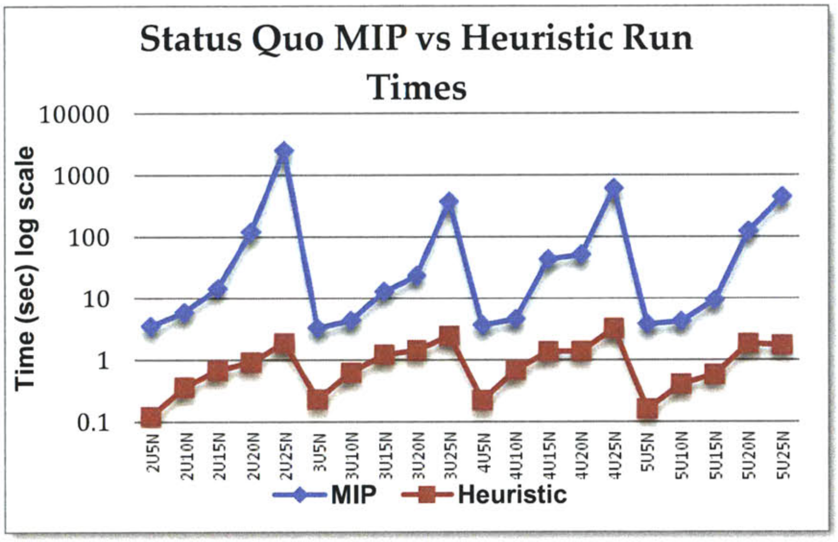


Figure 31 - Status Quo Concept MIP and Metaheuristic Run Times. The 2 UAS 25 Targets case had the largest run time difference (2472.96 sec for the MIP vs 1.85 sec for the metaheuristic with an optimality gap of .19%).

Tests	Avg Time (s)	CPU Time Diff	Mission Time (min)	Mission Time Diff	Wait Time (min)	Wait Time Diff	Obj Val	Optimality Gap
2U5N	0.298	-2.782	141.91	0	1.51	0	2225	0.00%
2U10N	1.37	-5.04	164.72	1	0	0	2685	0.00%
2U15N	1.53	-11.62	157.27	-4.15	9.08	9.08	4055	0.73%
2U20N	3.85	-122.81	169.82	2.88	23.03	21.28	5135	0.00%
2U25N	4.41	-1281.32	168.96	1.11	15.22	14.34	5220	0.00%
3U5N	0.82	-2.64	178.98	3	1.51	0	2220	0.00%
3U10N	2.13	-17.04	237.45	7.19	0	0	3210	0.00%
3U15N	2.5	-44.03	222.53	4.84	2.83	2.83	4315	0.00%
3U20N	2.45	-237.66	243.17	3	5.16	5.16	5215	0.00%
3U25N	5.33	-16126.98	243.97	2	5.16	5.16	5145	1.72%
4U5N	1.74	-8.24	176.09	16.23	0	0	2225	0.00%
4U10N	2.53	-55.92	281.86	16.9	0	0	3355	0.00%
4U15N	4.34	-4460.51	297.7	30.71	0	0	4465	0.00%
4U20N	4.71	*-410.26**	294.82	*	2.83	*	5385	1.10%
4U25N	6.97	*-5683.3**	324.1	*	2.83	*	5455	0.18%
5U5N	1.4	-95.15	137.17	9.41	0	0	2500	0.00%
5U10N	2.91	-1232.34	298.64	23.01	0	0	3430	0.00%
5U15N	5.69	*1.23**	331.28	*	0	0	4575	0.00%
5U20N	7.78	*-21.85**	375.64	*	6.09	*	5610	1.41%
5U25N	9.87	*-8227.13**	409.98	*	6.33	*	6570	2.95%

Table 6 – Swap and Share Metaheuristic Results and Comparison to Swap and Share MIP. The difference columns show the metaheuristic times minus the MIP times. The (*) annotates the results without the duty time minimization step explained in Section 4.1.2.1 due to insufficient memory. Mission Times did not deviate more than 9% from optimal mission time. Wait times increased for 6 out of the 16 test cases with known solutions.

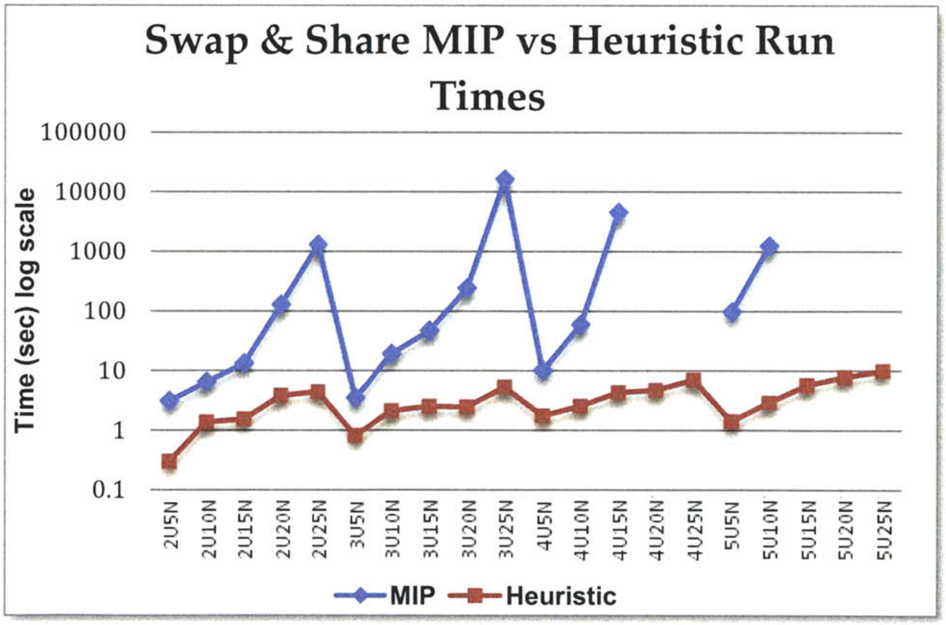


Figure 32 – Swap and Share Concept MIP and Metaheuristic Run Times. The 3 UAS 25 Targets case had the biggest run time difference (16132.31 sec for the MIP vs 5.33 for the metaheuristic with an optimality gap of 1.72%). The missing data points reflect the problem sizes that CPLEX could not solve for duty time minimization due to insufficient memory.

The results validate our fourth hypothesis in that the metaheuristics find the optimal objective function value in the majority of our test cases while dramatically reducing computation time. For the Status Quo concept, the metaheuristic found the optimal solution in 14 out of 20 test cases with an optimality gap of less than 1.77%. The largest time savings using the metaheuristic was 41.2 minutes with the 2 UAS 25 target case. For the Swap and Share concept, the metaheuristic also found the optimal solution in 14 out of 20 test cases with an optimality gap of less than 1.72% for the cases that were solved to optimality. The largest time savings for the cases that solved to optimality was 4.5 hours with the 3 UAS 25 target case.

5.2.4 Experiment 4A

Because unit commanders may be hesitant to swap Raven UASs with other adjacent units, a natural extension to this problem is to see what advantages, if any, are gained from swapping bases. For our next experiment, we test our fifth hypothesis to understand how collaborative sharing of targets for the Raven UASs with base restrictions affects our measures of performance. We used the same 20 pseudorealistic test sets, but limited the UASs to their respective bases. In order to restrict the swapping of bases, we reintroduce the following constraints from the Status Quo Concept into the Swap and Share Concept formulation for each Raven UAS Base:

(9) Balance Constraint for UASs

$$\sum_{i \in N} travel_{i,j,v} - \sum_{i \in N} travel_{j,i,v} = 0 \quad \forall j \in N, \forall v \in V$$

(10) Flow conservation constraint to ensure each Raven UAS leaves out of exactly one Raven base

$$\sum_{i \in R} \sum_{j \in N} travel_{i,j,v} \leq 1 \quad \forall v \in R$$

We consider 3 or more UASs because with 2 UASs, our test cases only consist of one Raven UAS and one Shadow TUAS and no sharing can occur under our original assumptions. We present our results in Figure 33.

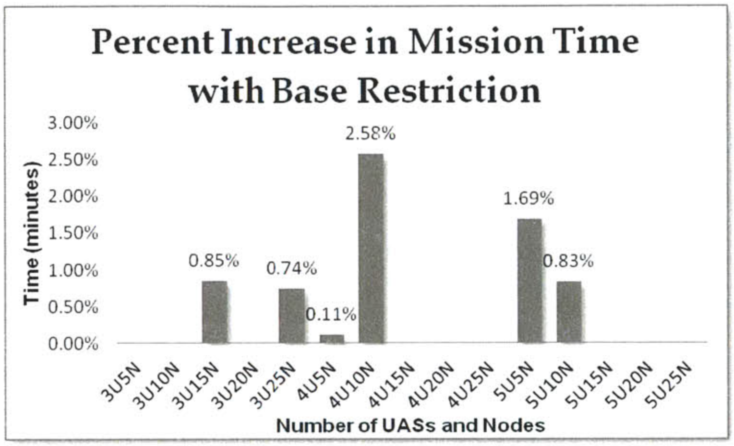


Figure 33 - Difference in Mission Time with Base Restriction when comparing with Swap and Share Concept. The only noticeable difference with the base restriction came with the mission time increase. Using the Swap and Share concept, the biggest difference was 6.84 minutes saved with swapping bases for 4 UASs 10 Targets. The missing data points reflect the problem sizes that CPLEX could not solve due to insufficient memory.

We validate our fifth hypothesis and find that each test case with and without the base restriction resulted in the same objective function value for the Swap and Share Concept. The only significant advantage relaxing the base restriction was a reduction in the total mission time, resulting from reduced travel time by allowing base swaps. The largest increase in mission time was a 2.58% increase involving 4 UASs and 10 Target nodes. We attribute these results to the Shadow TUAS's relatively faster speed and coverage ability in visiting the dual look targets, obviating the need to swap bases. We illustrate this rationale further with a simple worst case scenario to determine how swapping bases might prove advantageous in a given realistic scenario.

5.2.5 Experiment 4B

Suppose we have three UASs with their respective base locations as well as one primary mission for each UAS and two dual look LLO targets.

Furthermore, suppose that the Shadow TUAS's primary mission takes up the entirety of the horizon, preventing it from satisfying any of the dual look LLO targets. This possibility could exist especially if a surveillance directive from Division or higher is given. Alternatively, we can assume that the Shadow TUAS is not available during this planning horizon. In this case, we would predict that the Raven UASs, in an attempt to satisfy the dual look targets, would benefit from swapping bases for particular time window restrictions. We test a 3U5N test case, positioning the primary targets and dual look targets close in proximity to each Raven UAS base. By not allowing the swapping of bases, the objective value is 1660 because each UAS stays within its local area. By allowing the swapping of bases, the objective value is improved to 2000 because the Ravens can satisfy the dual look requirements for both of the dual look targets. The following figures illustrate this example further.

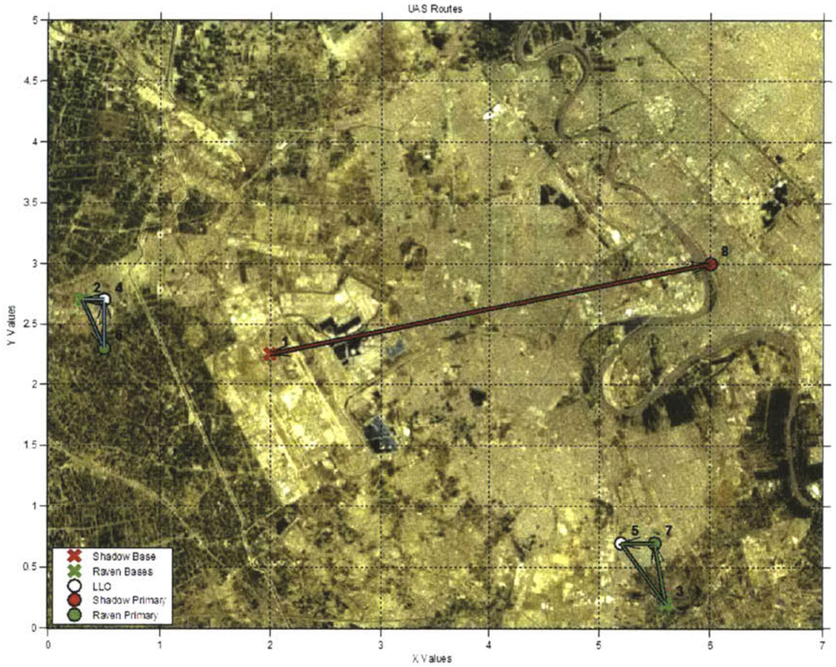


Figure 34 - 3 UAS 5 Targets Test Case Route Solution for Swap and Share with base restrictions. With the base restriction, each of the 2 Raven UASs can only observe its local targets despite the additional values available with the dual look LLO targets labeled in white.

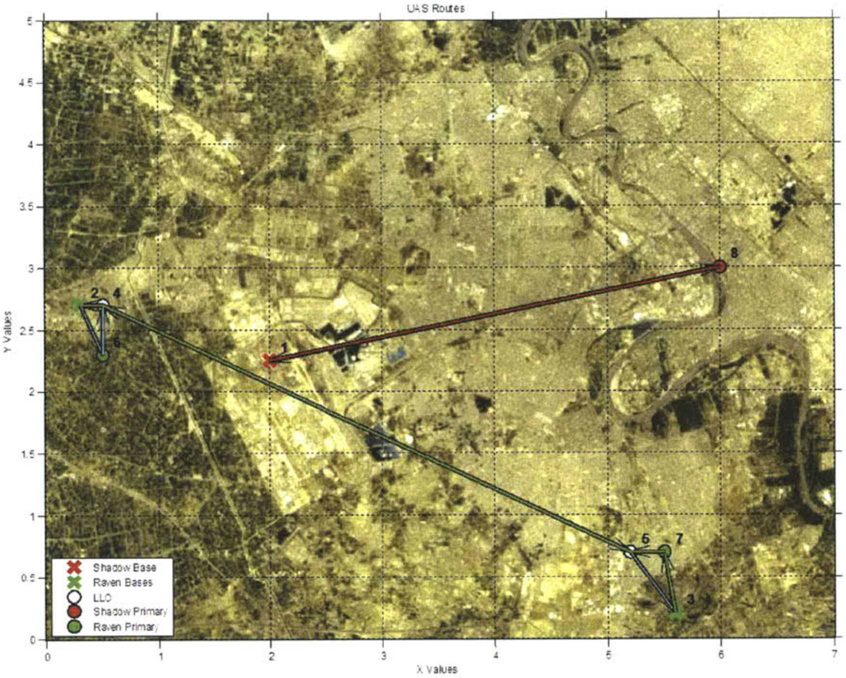


Figure 35 - 3 UAS 5 Targets Test Case Route Solution for Swap and Share with base restriction lifted. In this case, each of the 2 Raven UASs can observe both dual look targets in addition to its local targets. The two Raven routes overlap in this case

As observed with our 3 UAS 5 Target test case we expect to see added value with the base swap case as long as both of the following conditions are met:

1. The travel time to satisfy both dual look targets exceeds the UAS endurance for the base restriction case.
2. The travel time to satisfy both dual look targets is less than the endurance limit when base swaps are allowed.

If we assume minor changes in altitude, any scenario with localized targets near bases that satisfy the above conditions would result in an improved objective value by sharing bases.

5.2.6 Experiment 4C

We test a 4 UAS 6 Targets case to evaluate if the base restriction results above hold true for 3 Raven UASs. By not allowing the swapping of bases, the objective value is 2310 because each UAS must stay within its local area to satisfy time restrictions. By allowing the swapping of bases, the objective value is 2760, an improvement of 450 because the Ravens can collaboratively satisfy the dual look requirements for both of the dual look targets. The routes do follow the convex hull created by the raven bases as expected. The following figures illustrate this example further.

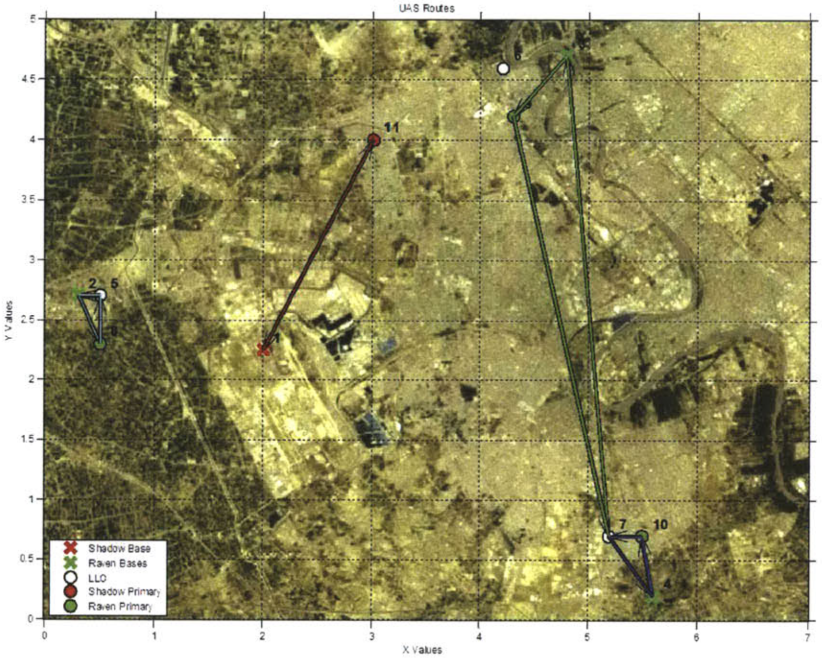


Figure 36 – 4 UAS 6 Targets Test Case Route Solution for Swap and Share Concept with base restrictions. With the base restrictions, each Raven UAS does its best to satisfy what it can reach. In this case, the northernmost Raven unit ignores the closest LLO target for the southern LLO target 7 in order to satisfy the dual look.

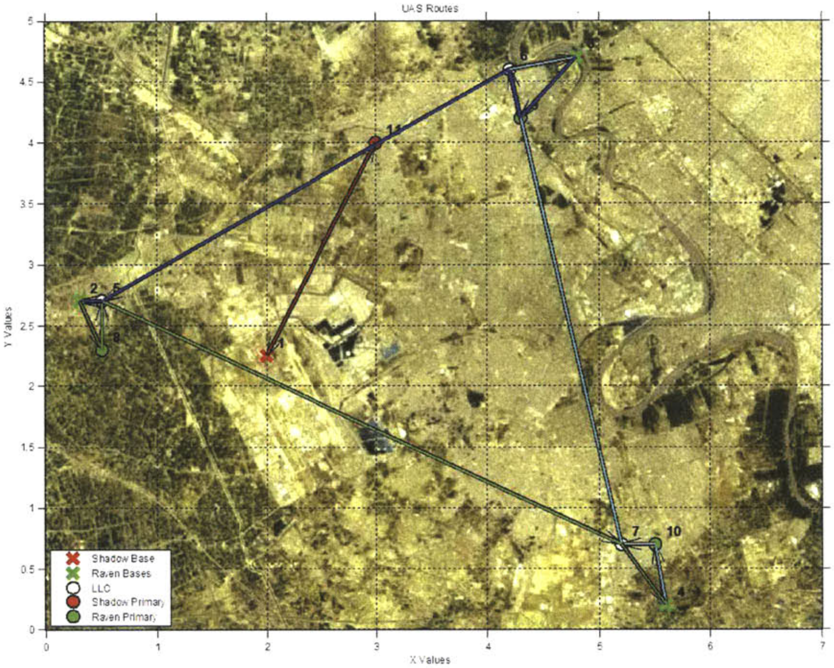


Figure 37 – 4 UAS 6 Targets Test Case Route Solution for Swap and Share Concept with base restrictions lifted. As expected, each of the 3 Raven UASs swaps bases in order to satisfy all of the dual look targets.

5.2.7 Experiment 4D

Lastly we test our conjecture that without the availability of the Shadow TUAS, the advantages of swapping bases are greater. We use one of our original pseudorealistic test cases and restrict the Shadow TUAS from observing any additional LLO targets while incrementally increasing the number of dual look nodes. We use the 4 UAS 20 Targets data set for tractability and adjust the parameter from 3 dual look nodes to 8 dual look nodes. By allowing base swapping, we find a case that provides a 10 point improvement to 4410 compared to 4400 for the base restriction scenario. This 10 point increase comes as a result of swapping bases, thereby allowing a Raven UAS to observe an LLO target node with a longer observation time of 15 minutes as opposed to one with a lower value and a 10 minute observation time.

Based on our fourth experiment, we reach the following conclusions:

1. The Swap and Share Concept with base restrictions results in lengthier mission times for the majority of cases. The concept does achieve the same objective function values in the majority of cases, confirming Hypothesis 5.
2. Without the availability of the Shadow TUAS, greater improvements to the objective value can be attained from swapping bases to satisfy dual look targets.
3. We can conclude with these supplementary experiments that scenarios exist when base swapping does provide a distinct advantage.

5.2.8 Experiment 5A

For our next set of experiments, we use random target sets to test the robustness of our Swap and Share metaheuristic. First, we randomly distribute targets and base locations as well as altitudes uniformly across the Brigade AO using Matlab's pseuodorandom generator. We stick to the 4 UAS case based on relevance and tractability and keep the number of dual look LLO targets constant at 3, with single look LLO targets proportional to 40% of the targets and primary targets making up the remaining nodes.

To provide a statistical measure for each base location relative to its respective primary targets and LLO targets, we determine the mean distance of the bases to each vertex of the convex hull line created by each UAS's primary targets and LLO targets using the following equation:

$$mean\ distance\ to\ convex\ hull\ vertices = \sum_{i=1}^{m} \frac{\sqrt{(x_i - x_b)^2 - (y_i - y_b)^2 - (z_i - z_b)^2}}{m}$$

where m represents the number of each UAS's primary targets plus the LLO targets, (x_i, y_i, z_i) represents the x and y coordinates and altitude of the convex hull vertices created by these nodes, and $(x_b, y_{b,} z_b)$ represents the base locations.

As another statistical measure, Principal Component Analysis (PCA) of the node locations can also identify patterns and reveal the internal structure that best explains the variance. To identify the principal components, we standardize the data composed of the x values, y values, and altitudes for each node and find the eigenvalues and eigenvectors of the covariance matrix. The eigenvector with the highest eigenvalue makes up the first principal component and accounts for the maximum amount of total variance of the three observed variables. The second principal component accounts for the amount of variance in the data set not accounted for by the first component. Appendix F shows the results of both statistical measures for each random data set.

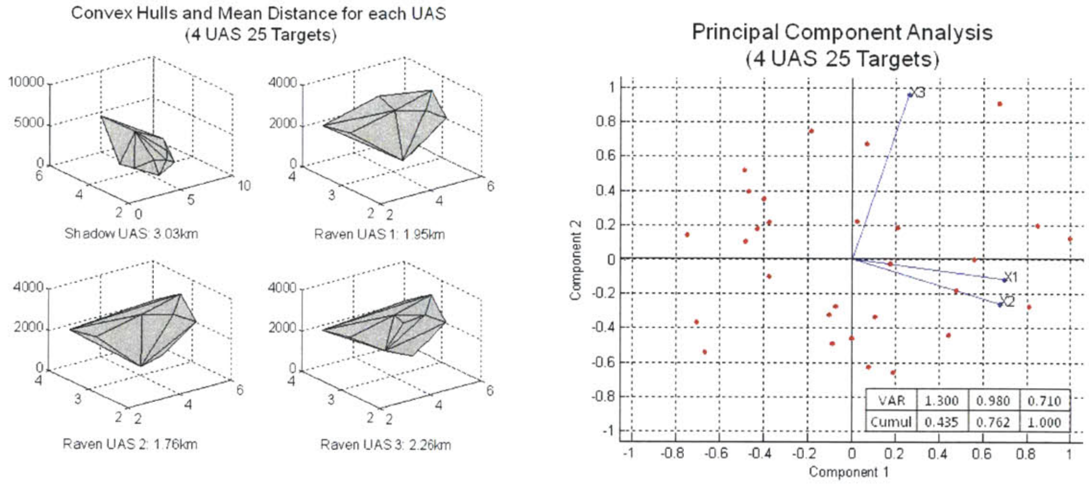


Figure 38 – Statistical Measures for Uniformly Distributed 4 UAS 25 random test set. Convex Hulls and Mean Distance to each vertex from UAS base locations are shown on the left and the Principal Components Analysis with the variance and cumulative variance of each component is shown on the right.

The results of the Swap and Share metaheuristic are shown in Table 7. The metaheuristic found the optimal solution in 3 out of 5 test cases with an optimality gap of less than .61% for the cases that could be solved to optimality. The largest wait time increase was 6.98 minutes or 3.11%. The largest run time savings for the cases that solved to optimality was 27.5 hours with the 4 UAS 25 target case.

Tests	Avg Time (s)	CPU Time Diff	Mission Time (min)	Mission Time Diff	Wait Time (min)	Wait Time Diff	Obj Val	Optimality Gap
4U5N	0.7	8.18	150.65	4.77	0	0	1539	0.00%
4U10N	2.24	15.94	271.45	-20.96	9.22	6.98	2597	0.61%
4U15N	4.76	540.67	300.36	15.33	0	0	3087	0.00%
4U20N	5.37	897.83	306.16	10.88	0.148	0.148	4096	0.00%
4U25N	4.85	99037.79	311.18	~	3.37	~	4572	0.54%

Table 7 -Swap and Share Metaheuristic Performance Results for randomized test sets. The largest optimality gap was .61% with uniformly randomized test sets. The (~) annotates the results without the duty time minimization step explained in Section 4.1.2.1 due to insufficient memory.

5.2.9 Experiment 5B

Furthermore, to test the limits of the metaheuristic, we observe how increasing the problem size impacts the run time performance. Based on the reasonable number of targets the Effects Working Group can monitor, we limit the maximum number of targets to 100. We steadily increase the number of dual look LLO targets by 4 and single look LLO targets by 6 with each incremental increase of 10 targets, while keeping the number of primary mission targets constant at 10. The run times for each problem size are detailed in Figure 39:

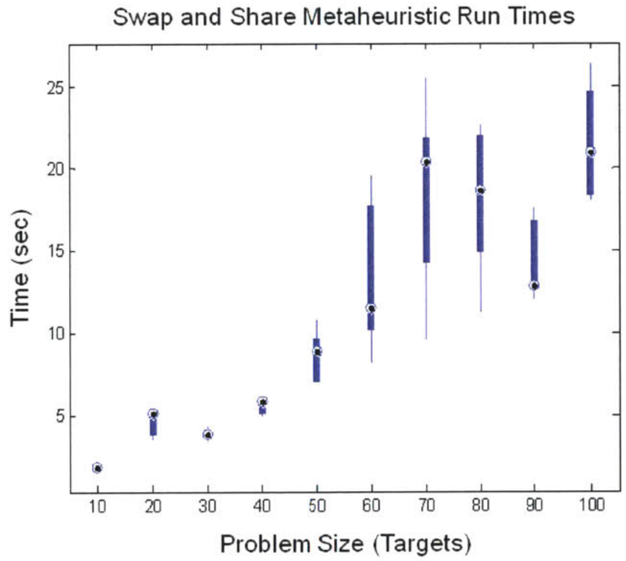


Figure 39 - Swap and Share Metaheuristic Run Times for Increasing Problem Size. This compact boxplot shows that the variance generally increases with the problem size.

The run time for the largest problem with 100 targets took an average of 21.5 seconds.

5.2.10 Experiment 5C

For a second random target set to test the metaheuristic's performance, we assess the benefits of swapping bases in terrain that necessitate linear route paths such as a border. We allocate a region of radius 3.6 km for each Raven Base to simulate the operating range. Because Raven units typically search targets within the operating range of the Raven, we distribute the LLOs and Raven Mission Targets within this circular region. Within this circular region, we generate pseudorandom numbers from a normal distribution and convert these into x and y coordinates using the following method:

1. Generate random base location to get the center points x_c and y_c
2. Generate two pseudorandom numbers from a normal distribution, x and y
3. Using the equation $r_b^2 = x^2 + y^2$, find r using $r = \sqrt{\frac{rand}{r_b}} \cdot radius$, where $rand$ is a pseudorandom number between 0 and 1, and $radius$ is the input.
4. Get the final x and y coordinates for the LLO:
   $x\ value = x_c + r \cdot x$
   $y\ value = y_c + r \cdot y$

Given the extended range of the Shadow TUAS, we randomly distribute its mission targets uniformly across the Brigade AO as before. Figure 40 shows an example test set with 20 target nodes.

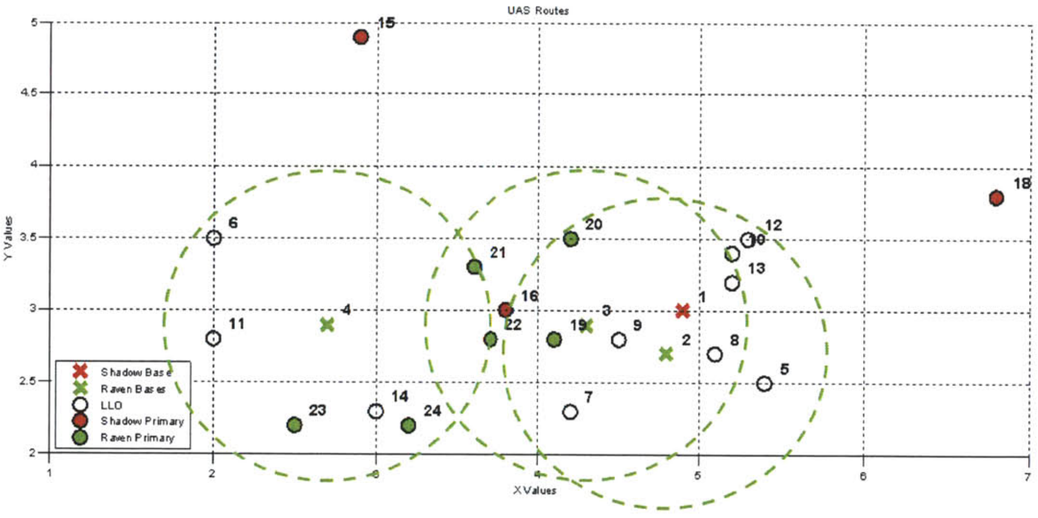


Figure 40 - Example random test set with linear characteristics. The green dashed circles represent the operating range for the Raven UAS. The Shadow TUAS range exceeds the given area explaining the two primary targets near the edges.

Using the same statistical measures as before, we compare the results of this random test set with the uniformly distributed random test set results summarized in Appendix F. For example, for the 4 UAS 25 target node case, the convex hulls are shown in Figure 41 along with the mean distance for each UAS. Compared to the randomly distributed targets scattered uniformly across the Brigade AO, the border test case will typically have larger mean distances due to the linear patterns influenced by the border. Additionally, the border test case will have smaller secondary principal components because more of the variance will be captured by the first component.

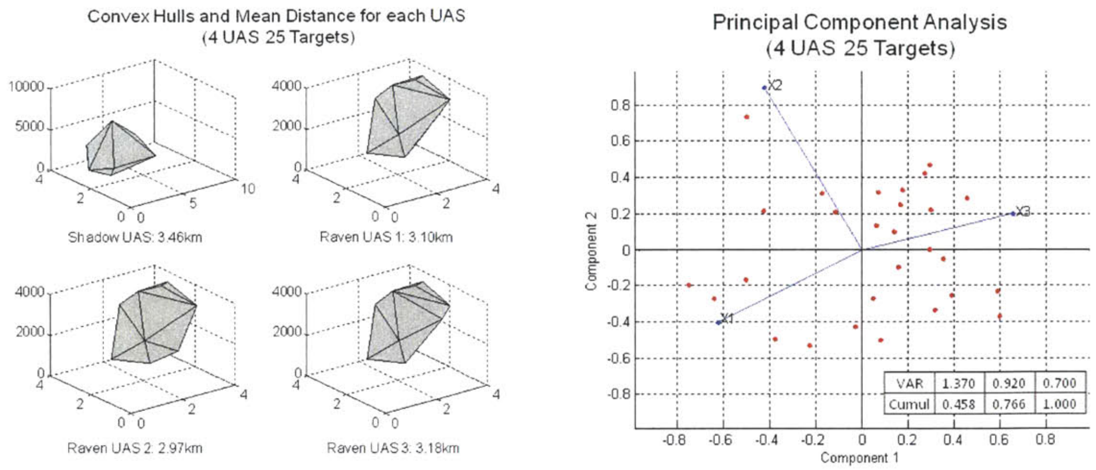


Figure 41 - Statistical Measures for Border Scenario 4 UAS 25 random test set. This random border test case will typically have larger mean distances due to the linear patterns influenced by the border and smaller secondary principal components because more of the variance will be captured by the first component.

Table 8 presents the metaheuristic performance results using the randomized border scenario.

Tests	Avg Time (s)	CPU Time Diff	Mission Time (min)	Mission Time Diff	Wait Time (min)	Wait Time Diff	Obj Val	Optimality Gap
4U5N	0.97	30.19	134.71	8.57	0	0	2034	0.00%
4U10N	3.26	204.32	308.1	7.79	0	0	3089	0.00%
4U15N	5.79	1495.25	304.29	-32.22	0	0	4090	0.46%
4U20N	9.25	70423.2	303.62	4.2	1.78	1.78	4068	0.46%
4U25N	15.34	20903.75	335.69	~	3.77	~	4607	0.00%

Table 8 -Swap and Share Metaheuristic Performance Results for randomized border scenario. The largest optimality gap was .46%. The (~) annotates the results without the duty time minimization step explained in Section 4.1.2.1 due to insufficient memory.

The metaheuristic found the optimal solution in 3 out of 5 test cases with an optimality gap of less than .46% for the cases that were solved to optimality. The largest wait time difference was 1.78 minutes and the largest time savings for the cases that solved to optimality was 19.5 hours with the 4 UAS 20 target case.

From experiment 5, we can substantiate our sixth hypothesis and conclude that our Swap and Share metaheuristic performs well with random target sets scattered both uniformly and in a more linear border scenario. The metaheuristic achieved objective values within a 1% optimality gap for the random target sets.

Table 9 provides a one look Chapter summary of the experiment results as well as performance differences with each concept covered.

Performance Measure	Status Quo	Swap and Share
Largest tractable problem size for MIP	5 UAS 25 targets	4 UAS 15 targets, 5 UAS 10 targets
MIP Run Times (increasing # of targets)	increases as a function of number of targets	increases as a function of number of targets
MIP Run Times (increasing # of UASs)	no pattern, can decrease with increasing UASs	increases as a function of number of UASs
MIP Objective Value	Total value for all test sets: 83005	Improvement with 6 of 20 cases for increase in value of 460, total value: 83465
Metaheuristic Performance	Biggest reduction for tractable problem: 41.2 min with optimality gap of .19%	Biggest reduction for tractable problem: 5.33 hours with optimality gap of 1.72%. Run time variance increases with problem size
Mission Time reduction		Results in decrease in mission time compared to base restriction concept, largest decrease 6.84 min, or 2.58%
Dual Look Increase effect on Swap and Share MIP run time		General increase. No pattern in reaching MIP exact solution, can decrease with more dual look targets.
Dual Look Increase effect on Swap and Share Metaheuristic run time		Minimal impact on run time. Biggest reduction in run time compared with MIP: 34.3 min, Largest optimality gap: 0.33%
Metaheuristic performance on uniform random test set vs MIP		Biggest run time reduction: 27.5 hours compared to MIP. Largest optimality gap: 0.61%
Metaheuristic performance on random border scenario vs MIP		Biggest run time reduction: 19.5 hours compared to MIP. Largest optimality gap: 0.46%

Table 9 -Experiment and Performance Measure Results

6 String Extension to Swap and Share Concept

Based on the results and the established utility of the Swap and Share CONOP, this section explores an extension of the model to include multiple sorties for each given time horizon. This extension not only allows for a more accurate reflection of current operations, it also simplifies the problem formulation by the use of composite strings that encompass all of the constraints introduced in the earlier model. A metaheuristic adapted from the previous model creates feasible strings that detail the multiple sorties each UAS can execute during the planning horizon. Barnhart et al. [2] used strings to model flights for aircraft fleeting and routing. The authors show the robustness of such an approach especially in dealing with complicated constraints like maintenance and aircraft utilization restrictions. Similar to how the authors defined a string, a string in our model is defined as a sequence of connected flights that begin and end at a UAS base location and satisfies flow balance. For example, a string can include multiple sorties by a UAS within a given time horizon.

As outlined in Section 2.3.3, the S2 Collection Manager creates an ISR Synchronization Matrix as one of three tools to assist in dissemination of information and analysis. More often than not, this synchronization matrix will include multiple sorties for each UAS over a given timeframe based on SIR requirements. In Figure 42, the S2 Collection Manager's ISR Synchronization Matrix shows each UAS and its primary target with line segment nodes representing ETIOV and LTIOV and flexible observation time requirements in horizontal brackets.

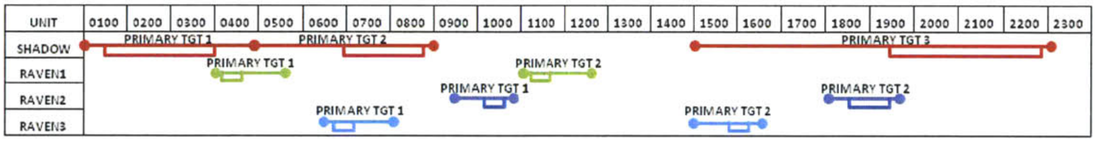


Figure 42 – ISR Synch Matrix Example. This example shows how an ISR synch matrix would look as a product of the ISR Synchronization process detailed in section 2.1.8.

Equipped with an LLO target requirement from the Effects Working Group, the S2 CM must then insert these LLOs with their own associated observation times, ETIOV, and LTIOV. An example of the final augmented ISR Synch Matrix is shown in the last timeline. It is important to note that Raven UASs, with a 90 minute endurance, must return to base following each primary mission requirement. Additionally, because Raven units work collaboratively by sharing primary targets and bases, the resulting plan deviates significantly from the original schedule.

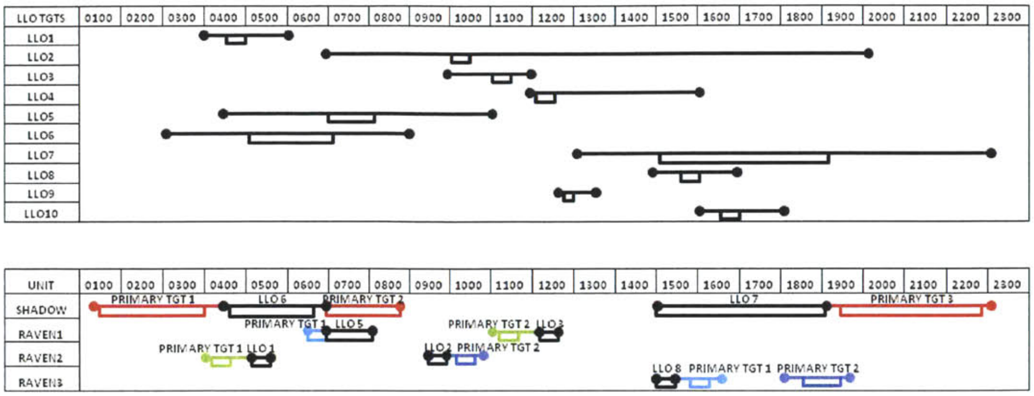


Figure 43 – ISR Synch Matrix with LLO. The top matrix shows the candidate LLOs from the Effects Working Group. The bottom matrix shows the final augmented ISR Synch Matrix after LLOs are integrated into the original plan.

6.1 String Concept Development

6.1.1 String Concept Heuristic

The String heuristic initially inserts base stops in to the initial routes based on the UAS endurance limits and the ETIOV's of the primary target nodes. This allows for the decomposition of the UAS's route string for the entire horizon into individual sorties. Furthermore, this also ensures that the number of sorties stays consistent with the original plan set forth in the ISR Synchronization Matrix. This step essentially creates the sorties observed in Figure 42. The decomposition algorithm can be summarized with the following pseudocode:

```
for each UAS $v$
    Sort Primary targets for UAS $v$ in ascending order and insert primary
        target if feasible
    Get BestRoute
    Initialize $num\_subroute = 0,\ position\ (m) = 1$
    while $m <$ length of BestRoute
            if $BestRoute(m+1).ETIOV + BestRoute(m+1).obstime +$
               $traveltime(m+1, v, v) - UAS.endurance * $
               $num\_subroute > UAS.endurance$ AND
```

```
            $BestRoute(m).ETIOV + BestRoute(m+1).obstime + traveltime(m, v, v) - UAS.endurance * num_{subroute} \le UAS.endurance$
                Insert UAS v base at position $m + 1$
                $num\_subroute = num\_subroute + 1$
            end if
            $m = m + 1$
        end while
end for
```

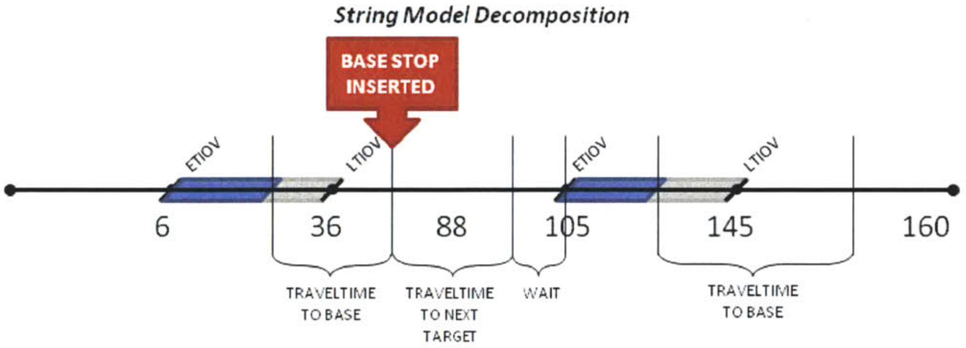


Figure 44 - String Model Decomposition. In this example, a base stop is inserted based on the algorithm that identifies when a return to base is required based on a conservative assessment of the UAS endurance. The travel time from the base to the next target will always be sufficient based on the original ISR Synch Matrix.

After insertion of the base stops, the metaheuristic for the String Concept first applies the insertion heuristic to each of these decomposed sorties, to accurately account for the available single and dual look LLO targets with each step.

The following 2 opt intra-route, Deletion-Insertion inter-route, and the 2 Exchange inter-route heuristics take the entire string through the modification first and then decomposes the resulting string into sorties to check feasibility, delineating with base stops. For example, suppose a string with two sorties looks like [1, 19, 20, 4, 1, 8, 21, 1], with the base at node 1. The 2 opt heuristic might swap the third and fifth element of the route resulting in a new route that looks like [1, 19, 1, 4, 20, 8, 21, 1]. This new route can be decomposed into two subroutes [1, 19, 1] and [1, 4, 20, 8, 21, 1] to check for feasibility and for the shortest travel time.

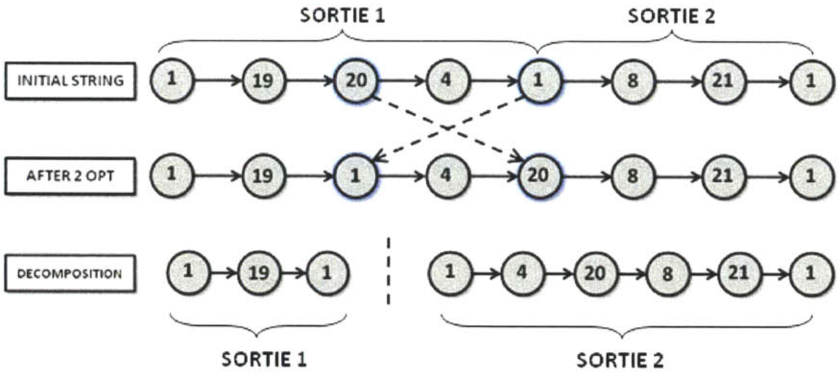


Figure 45 –String Model Decomposition after improvement step. Here the string is the multiple sortie route [1,19,20,4,1,8,21,1]. The two new routes created from the decomposition are checked for feasibility.

As a final step, the String Metaheuristic applies a second iteration of the insertion heuristic. The String Metaheuristic creates feasible strings that provide routes for each UAS, within the constraints of the problem. For example, feasible strings for a simple 3 UAS problem might look like the following:

$$\begin{pmatrix} [1\ \mathbf{7}\ 5\ \mathbf{8}\ 1] \\ [1\ 5\ \mathbf{7}\ 1\ \mathbf{8}\ 1] \\ [1\ \mathbf{7}\ 5\ \mathbf{8}\ 1\ 6\ 1] \\ [1\ \mathbf{7}\ 5\ \mathbf{8}\ 1\ 6\ 4\ 1] \\ [2\ 4\ \mathbf{9}\ 2] \\ [2\ 6\ 2\ 4\ \mathbf{9}\ 2] \\ [2\ 5\ \mathbf{9}\ 2] \\ [3\ \mathbf{10}\ 5\ 3\ 6\ 3] \\ [3\ 5\ \mathbf{10}\ 3\ 6\ 3] \\ [3\ \mathbf{10}\ 5\ 3] \end{pmatrix}$$

Here, there are 4 feasible strings for UAS 1, 3 strings for UAS 2, 3 strings for UAS 3, some with multiple sorties. The bold numbers represent the mission targets and the dual look nodes in this example are nodes 4, 5, and 6. The String Metaheuristic arrives at these unique strings by developing feasible string sets, which is defined as a set of combined routes for each individual UAS that meet problem constraints. For example, a feasible string set for our above example would be:

$$\begin{pmatrix} [1\ \mathbf{7}\ 5\ \mathbf{8}\ 1\ 6\ 4\ 1 \\ [2\ 4\ \mathbf{9}\ 2] \\ [3\ \mathbf{10}\ 5\ 3\ 6\ 3] \end{pmatrix}$$

From the feasible strings, we calculate objective values to use as inputs for the integer program described in the next section.

6.1.2 Integer Programming Problem

Using the aircraft maintenance routing feasibility problem described in Chapter 3 as a basis for our string model extension, we use the following formulation:

Sets:

N: Set of all nodes
N_s: Set of all single look LLO nodes
N_d: Set of all dual look LLO nodes
S: Set of all feasible strings
K: Set of bases (equal to the number of UASs)
S_K^+: Set of strings originating at station K
S_K^-: Set of strings terminating at station K

Inputs:

$LLOsingle_{is}$: 1 if single LLO mission node i is included in string s, 0 otherwise
$LLOdual_{is}$: 1 if dual LLO mission node i is included in string s, 0 otherwise
$dvalue_s$: Additional dual look reward value for string s
$value_s$: Reward value for string s
$wait_s$: Total wait value for string s

Decision Variables

x_s: 1 if string s is included in solution, 0 otherwise

(38) Objective Function

$$maximize \sum_{s \in S} \left(value_s + \frac{dvalue_s}{2} \right) \cdot x_s$$

Constraints

(39) Constraint to ensure that all single look LLO nodes are covered at most once

$$\sum_{s \in S} LLOsingle_{is} x_s \leq 1 \quad \forall i \in N_s$$

(40) Constraint to ensure that all dual look LLO nodes are covered at most twice

$$\sum_{s \in S} LLOdual_{is} x_s \leq 2 \quad \forall i \in N_d$$

(41) Aircraft balance at bases

$$\sum_{s \in S_K^+} x_s - \sum_{s \in S_K^-} x_s = 0 \quad \forall k \in K$$

(42) Aircraft count

$$\sum_{s \in S} x_s \leq K$$

(43) Binary constraint for decision variable x_s

$x_s \in \{0,1\} \quad \forall s \in S$

Here we take a simplified approach to handle the dual look node constraints, allowing UASs to achieve half of the dual look node value even if the node is only observed once. We solve this binary integer program within Matlab as a final step after creation of the strings. In matrix form, the inputs to the $Ax = b$ integer program includes an m x n concatenated matrix A as described in Appendix E where m is the number of LLOs plus 2 times the number of UASs, and n is the number of strings, and a b vector satisfying the right hand side of each constraint.

The String Concept's Metaheuristic can be summarized with Figure 46.

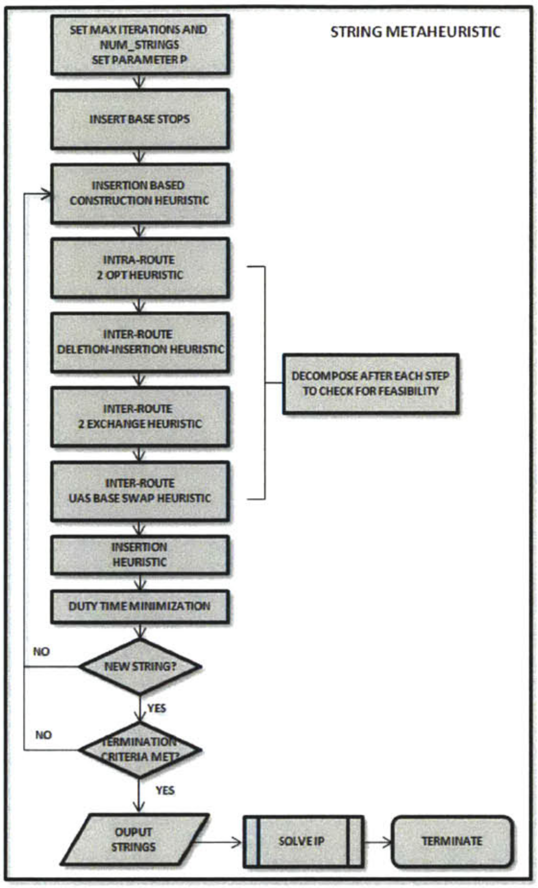


Figure 46 - String Concept Metaheuristic. The generations of unique strings become the input for the IP.

6.1.3 Experiment 6

As a final experiment, we create a uniformly distributed random test set similar to Figure 42's ISR Synch Matrix Example to replicate realistic mission sets for multiple UASs along with secondary LLO requirements. We test 3, 4, and 5 UAS cases with 20, 25, and 30 target nodes with 2 to 3 sorties for each UAS. As parameter inputs, we set the maximum number of string sets to 100, maximum iterations with no improvement to 50, *p* for the repeated local search mentioned in Section 4.2.3.4 to 0.6, and neighbors *n* mentioned in Section 4.2.3.1 to 1. To provide an example of the optimized Matlab output, the string metaheuristic with a 3 UAS 15 target node case results in an augmented UAS schedule as shown in Figure 47:

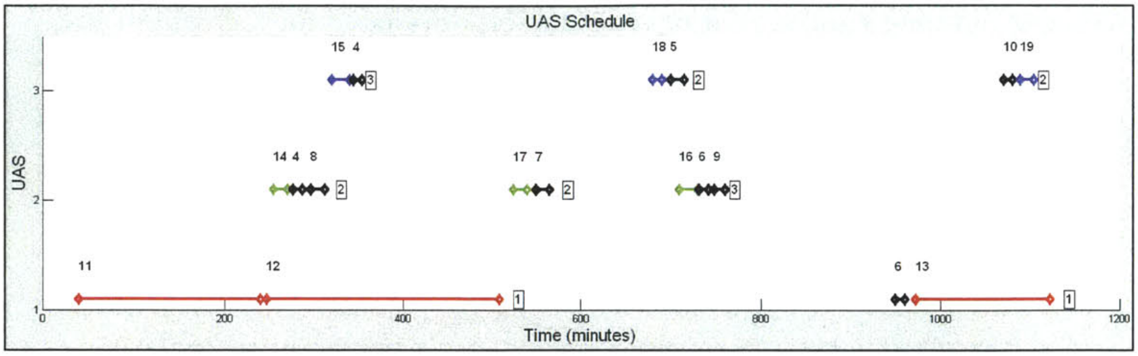


Figure 47 – Resulting String Concept Schedule for 3 UAS 15 target nodes. In this case, Raven UAS 2, highlighted in green (second row), and Raven UAS 3, highlighted in blue (first row), swap bases. LLO targets are labeled in black and the intermediate base arrivals are shown in boxed numbers. For example, for Raven UAS 2, the final route string is [2,14,4,8,2,17,7,2,16,6,9,3]

The results of the test cases are shown in Table 10 with the number of total unique strings developed, the objective value, and the average total run time, mission time, and wait time. As expected, the number of unique strings developed, objective function value, and run time generally increases with the number of targets as the heuristic searches more of the solution space. The 4 UAS 25 target node case took the longest time to solve at 3.28 minutes most likely due to the favorable spatial proximity of random node locations and times that created more feasible combinations.

Tests	num strings	time (sec)	mission time (min)	wait time (min)	Obj Val
3U20N	26	37.29	970.63	75.43	6095
3U25N	39	39.67	981.69	87.84	6260
3U30N	126	73.09	1227.32	75.2	7940
4U20N	25	27.47	886.98	77.69	5425
4U25N	159	196.85	1183.71	169.38	9115
4U30N	182	96.76	1270.72	97.63	9685
5U20N	50	54.87	62.64	958.61	7340
5U25N	209	159.87	1284.21	198.04	10800
5U30N	217	93.45	1633.04	330.54	11875

Table 10 –Swap and Share String Concept Results for randomized multiple sortie scenarios. As expected, the number of unique feasible strings developed increases as a function of the number of targets. The time shown includes the time to generate strings as well as the time to solve the IP.

In order to determine the ideal parameter setting for the maximum number of string sets to develop, we look at the change in the objective function achieved with the number of unique string sets developed for each of our test sets.

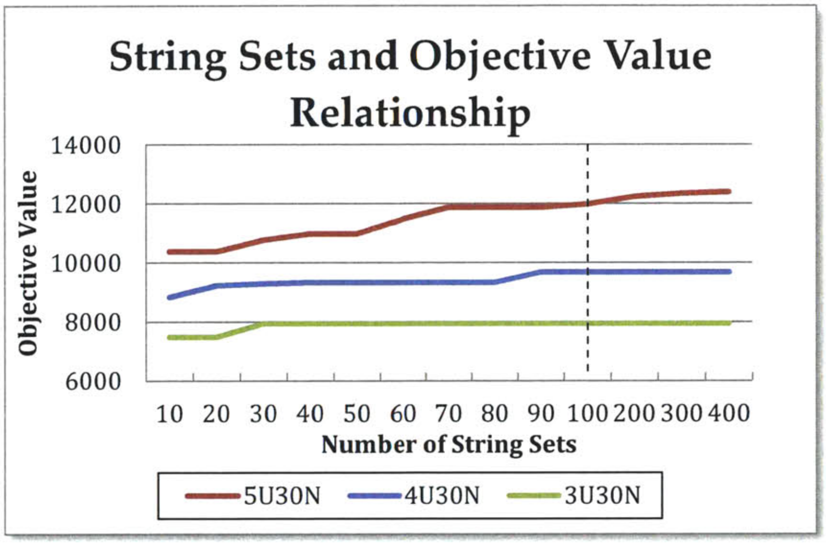


Figure 48 - String Sets and Objective Value Relationship. For a lower number of UASs, increasing the maximum number of string sets generated does not contribute to improved objective values. For 5 UASs, increasing the number of string sets to 400 achieved the maximum objective function value although it took 27.6 minutes of run time.

From Figure 48 we can conclude that the number of UASs in the problem and time constraints should dictate the parameter setting for the maximum number of string sets generated. In our largest test case involving 5 UASs and 30 target nodes, setting the parameter to 100 reasonably solved the problem in 1.56 minutes and achieved an objective value within 4% of the maximum optimized value attained. The output for this problem in Figure 49 shows the integration of a significant number of secondary LLO targets and 2 of 3 dual look targets.

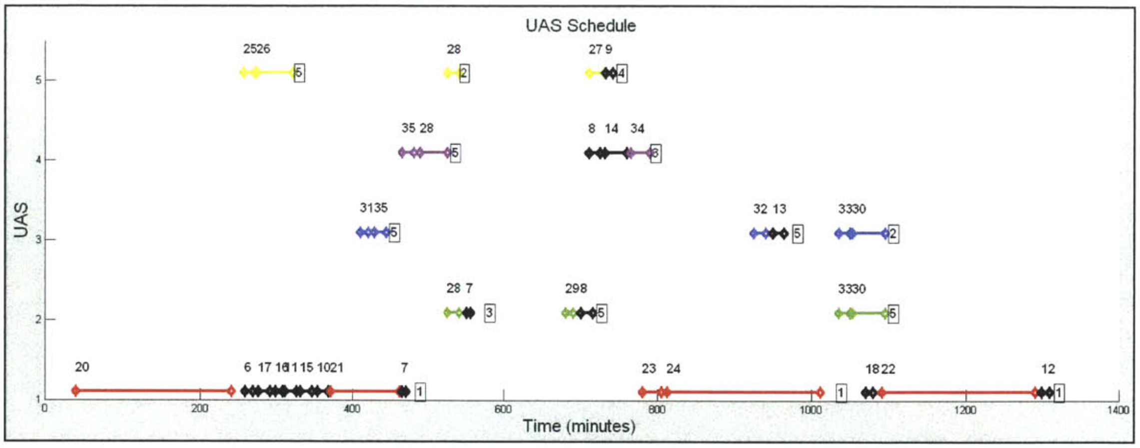


Figure 49 – String Concept Result Schedule for 5 UAS 30 target nodes.

The string model extension and its metaheuristic allows for optimization based planning and flexibility especially in accounting for human input. For example, the S2 Collection Manager could include additional operational constraints like a target's requirement for a specific type of UAS in the generation of strings. While the Swap and Share metaheuristic can only be applied to single sortie missions, this string metaheuristic allows for an efficient optimization of multiple sortie plans for each UAS by incorporating the constraints of the problem in the generation of strings. Each candidate string incorporates both LLO targets as well as the benefits of collaborative sharing of targets and bases to provide an overall improved plan ready for units to execute. In application, Brigades can incorporate this metaheuristic in their daily battle rhythms after development of an ISR Synch matrix along with separate candidate LLO requirements developed by the EWG. Furthermore, the few minutes it takes to reach an optimized plan fits well within the time constraints for mission planning.

[This Page Intentionally Left Blank]

7 Conclusions and Future Work

This chapter reviews the results of our experiments, summarizes the contributions of this thesis, and discusses how to advance the metaheuristic for the string concept. We also discuss future applications of dynamic vehicle routing and randomness in observation times.

7.1 Summary of Contributions

As stated in Chapter 1, the goal of this thesis was to enhance a given initial multi-route schedule for a diverse set of UASs by maximizing the observation of a diverse set of COIN targets. For this problem called the Collective UAS Planning Problem (CUPP), we aimed to present a new paradigm, termed the Swap and Share Concept, for a collective sharing of Raven UAS targets and base facilities. We conclude that the value of the swap and share concept lies not only in the increased observation of targets but also in reducing overall mission time. In addition, the Swap and Share concept allows for the maximization of limited resources as shown from experiments when limiting the Shadow TUAS. Higher echelon requirements and unforeseen maintenance requirements can reduce the number of UASs and supporting personnel available for missions at any time. After confirming the utility of the Swap and Share concept, we developed a metaheuristic to generate multiple sorties using our String concept to create feasible multiple sortie plans using decomposition. By solving sensible sized problems in a short period of time, units can seamlessly integrate this planning tool into their ISR Synchronization Process as described in section 2.3 to maximize use of UASs in a counterinsurgency. This thesis makes the following contributions:

1. A mixed integer programming formulation for the CUPP using both the Status Quo and Swap and Share concepts implemented in IBM ILOG OPL.
2. The development of metaheuristics implemented in Matlab for both concepts to solve the CUPP for realistic problem sizes of up to 5 UASs and 25 target nodes with an optimality gap of less than 1.72% and an average reduction in run time of 92%.
3. The development of a string model extension to solve for multiple sorties using metaheuristics and integer programming implemented in Matlab.
4. The development and experimentation of pseudorandom and pseudorealistic test sets.
5. Computational studies and results from applying both concepts along with the string model extension for realistic scenarios requiring tuning of parameters. The largest test case involving 5 UASs and 30 target nodes with 100 string sets generated solved the problem in 1.56 minutes with an objective value within 4% of the maximum optimized value attained.
6. Recommendations for modifications and for future work related to the CUPP.

7.2 Future Work

7.2.1 Additional Types of Targets

While our formulation covered optional single and dual look targets as well as mandatory primary mission targets, a modification to our problem would integrate semi-persistent targets. Especially in a COIN environment, semi-persistent targets entail those targets that hold some value for each visit, with no limit to what asset observes it or how many times an asset observes it. For example, if an intelligence report referenced a possible sabotage attack on a power plant with no clear timeframe, the Effects Working Group might request this as a semi-persistent target.

7.2.2 Conditional Swapping of Bases

Our string based concept currently swaps bases if the resulting route string decreases total travel time. However, in order to minimize the amount of swapping that occurs, an upgraded heuristic would only swap if it proves advantageous for inserting an additional target either with the current sortie or the one immediately following the base visit. This requires looking ahead to gauge if subsequent sorties with the base swap satisfy new target insertions; this additional step will increase run time and complexity.

7.2.3 Immediate Pop-Up Targets

A more realistic model for our problem would incorporate the concepts of the dynamic vehicle routing problem in which information can change after the initial routes are constructed, especially with regards to immediate pop up targets. These immediate targets might appear in real time during execution of the initial route plan. As discussed in section 2.3.4, dynamic retasking targets include top priority support for troops in contact with the enemy or time sensitive targets. As with the dynamic vehicle routing problem, this new model must always account for vehicle locations at all times, especially when immediate target requests occur based on some probability distribution. A hypothesis might ask if allowing for Raven UASs to swap bases with no net loss of Ravens and to share targets will not only allow for an increased objective value but also allow for an increased degree of dynamism. The degree of dynamism entails visiting immediate pop up priority targets with minimal wait time during mission execution. To measure how dynamic the routing system is, we use the 'Effective degree of dynamism' introduced by Larsen et al. [23], denoted by $edod_{tw}$, which incorporates reaction time. The reaction time of the ith request is denoted by $r_i = l_i - t_i$ where l_i is the latest possible time at which the service can begin and t_i is the time the request of the dynamic customer is received.

$$edod_{tw} = \frac{1}{n_{tot}} \sum_{i=1}^{n_{imm}} \left(1 - \frac{r_i}{T}\right)$$

Where T represents the planning horizon, n_{imm} denotes the number of immediate customers, and n_{tot} denotes the total number of customers.

With this dynamic optimization problem, the distribution of the slack or waiting time at each node influences the reaction time of the UAS to the immediate pop up target. A priori knowledge of common threat areas can help allocate slack in an optimal manner, influencing the vehicle location and thus allowing more flexibility to react to the immediate pop up request. J. Branke et al. [3] demonstrate that a good waiting strategy can significantly reduce the average travel time to serve the new customer.

7.2.4 Random Service Times

Another realistic extension to our model would involve random service times because observation times might change once the UAS arrives on target. For example, targets that only require visual confirmation might only take a second or two to complete instead of the requested ten minutes of coverage. This would likely lead to the inclusion of additional targets and hence an increased objective value. Simulation could be used to see the effects of random service times on the overall objective value, and these times could be modeled with a uniform probability distribution. O'Rourke et al. [27] assumes that the actual service time of target i, S_i falls between the minimum service time $S_{min}(i)$ and the maximum service time $S_{max}(i)$ under the following probabilities:

$$S_i = \begin{cases} S_{min}(i), & with\, probability\ .7 \\ Uniform(S_{min}(i), S_{max}(i)), & with\, probability\ .3 \end{cases}$$

A similar distribution could be used to model random service times.

Additionally, with random service times, the concept of soft time windows could also be applied based on the characteristics of many of the LLO targets. For example, with regards to the LLO target that requires observation of the Host Nation's security force training, observation during the planned training period would provide maximum value while deviations outside the training period would result in lesser value. With targets like this, these deviations might include observations of the setup of the training mission or tear down activity, both of which could provide some value to measure the effectiveness of the unit. For example, Tavakkoli-Moghaddam et al. [33] uses soft time windows for customer i who has a high satisfaction time window [ai, bi] and a hard time window [LBi, UBi] where LBi < ai and UBi > bi. In the intervals [LBi, ai) and (bi, UBi], service can occur but with a set penalty.

[This Page Intentionally Left Blank]

References

[1] Archetti, C., Hertz, A., and Speranza, M. "Metaheuristics for the team orienteering problem." *Journal of Heuristics*, 13:49-76, 2007.

[2] Barnhart, C., Boland, N. L, Clarke, L. W., Johnson, E. L., Nemhauser, G. L., and Shenoi, R. G. "Flight String Models for Aircraft Fleeting and Routing." *Transportation Science* 32, 3 (1998): 208-220.

[3] Branke, J., Middendorf, M., Noeth, G., and Dessouky, M., "Waiting Strategies for Dynamic Vehicle Routing."

[4] Chao, I-M., Golden, B., and Wasil E. A. (1996): "The team orienteering problem". *European Journal of Operational Research*, 88, 101–111.

[5] CW2 Lefebvre, C. (personal communication, 1 April 2011).

[6] Feillet, D., Dejax, P., and Gendreau, M. "Traveling Salesman Problem with Profits," *Transportation Science* Vol. 39, No. 2 May 2005, 188-205, ISSN 0041-1655.

[7] Field Manual Draft 2-01. (March 2009). *Intelligence, Surveillance, and Reconnaissance (ISR) Synchronization.* Washington, D.C.: Headquarters, Department of the Army.

[8] Field Manual 2-19.4. (November 2008). *Brigade Combat Team Intelligence Operations.* Washington, D.C.: Headquarters, Department of the Army.

[9] Field Manual 3-24. (December 2006). *Counterinsurgency.* Washington, D.C.: Headquarters, Department of the Army.

[10] Field Manual 3.34.170. (March 2008). *Engineer Reconnaissance.* Washington, D.C.: Headquarters, Department of the Army.

[11] Field Manual 34-8-2. (May 1998). *Intelligence Officer's Handbook.* Washington, D.C.: Headquarters, Department of the Army.

[12] Field Manual 6-0. (September 2011). *Mission Command.* Washington, D.C.: Headquarters, Department of the Army.

[13] Garcia, A., Vansteenwegen, P., Souffriau, W., Arbelaitz, O., and Linaza, M. (2009). "Solving Multi Constrained Team orienteering Problems to Generate Tourist Routes."

[14] Gendreau, M. and Potvin, J-Y. editors, Handbook of Metaheuristics, 2nd edition, chapter 3, Springer, 2010.

[15] Golden, B. L., Levy, L., and Vohra, R. "The Orienteering Problem," *Naval Research Logistics,* vol. 34, 1987, pp. 307-318.

[16] Guvendiren, Z., Downey, S., "Putting the Priority Back Into PIR: PIR Development in a COIN Environment." *Small War Journal* 2009.

[17] Held, M. and Karp, R. M. (1962), "A Dynamic Programming Approach to Sequencing Problems", *Journal of the Society for Industrial and Applied Mathematics* 10 (1): 196–210.

[18] "Irregular Warfare (IW) Joint Operating Concept (JOC)", Version 1.0, United States Department of Defense, 27 February 2009

[19] Jones, D, *Understanding Measures of Effectiveness in Counterinsurgency Operations*. May 2006.

[20] Larson, R., and Odoni, A. (1981). "Urban Operations Research." Englewood Cliffs, NJ: Prentice Hall.

[21] Kallehauge, B. et al. (2008), "Formulations and exact algorithms for the vehicle routing problem with time windows," *Computers and Operations Research*. Vol 35, no 7, pp 2307-2330.

[22] Kilcullen, David (28 September 2006). "Three Pillars of Counterinsurgency" Lecture, Counterinsurgency Conference, Washington DC.

[23] Larsen, A. (2000). "The dynamic vehicle routing problem." Ph.D. thesis, Technical University of Denmark, Kongens, Lyngby, Denmark.

[24] Miller, J. (2007). "Large-Scale Dynamic Observation Planning for Unmanned Surface Vessels." Masters Thesis. Operations Research Department. Massachusetts Institute of Technology, Cambridge, MA.

[25] Negron, B. (2009). "Operational Planning for Multiple Heterogeneous Unmanned Aerial Vehicles in Three Dimensions." Masters Thesis.

Operations Research Department. Massachusetts Institute of Technology, Cambridge, MA.

[26] Nikolaev, A. G., and Jacobson, S. H. "Simulated Annealing", *Handbook of Metaheuristics* (M. Gendreau and J.-Y. Potvins, Editors), Second Edition, Chapter 1, International Series in Operations Research and Management Science, Volume 146, Springer, New York, 1-40, 2010

[27] O'Rourke, K.P., Bailey, T., Hill, R., and Carlton, W. "Dynamic Routing of Unmanned Aerial Vehicles Using Reactive Tabu Search," AFIT/GOA/ENS/99M-06. School of Engineering, Air Force Institute of Technology, Wright Patterson AFB OH, December 1999.

[28] Righini, G., Salani, M. "Decremental state space relaxation strategies and initialization heuristics for solving the Orienteering Problem with Time Windows with dynamic programming," *Comput. Oper. Res.* 36 (4) (2009) 1191–1203, ISSN 0305-0548.

[29] Righini, G., Salani, M. "Dynamic programming for the orienteering problem with time windows." Technical Report 91, Dipartimento di Tecnologie dell Informazione, Universita degli Studi Milano, Crema, Italy, 2006.

[30] Ropke S, Cordeau JF, Laporte G (2007) "Models and branch-and-cut algorithms for pickup and delivery problems with time windows." Networks 49:258–272

[31] Schisler, S. (personal communication, 19 January 2012).

[32] Tang, H., & Miller-Hooks, E. (2005). "A tabu search heuristic for the team orienteering problem." *Computers & Operations Research*, 32, 1379-1407.

[33] Tavakkoli-Moghaddam, R., Safaei, N., Shariat, M., "A Multi-criteria vehicle routing problem with soft time windows by simulated annealing." *Journal of Industrial Engineering International Islamic Azad University*, Tehran South Branch. September 2005, Vol. 1, No. 1, 28–36.

[34] Training Circular 2-01. (September 2010). *Intelligence, Surveillance, and Reconnaissance Synchronization.* Washington, D.C.: Headquarters, Department of the Army.

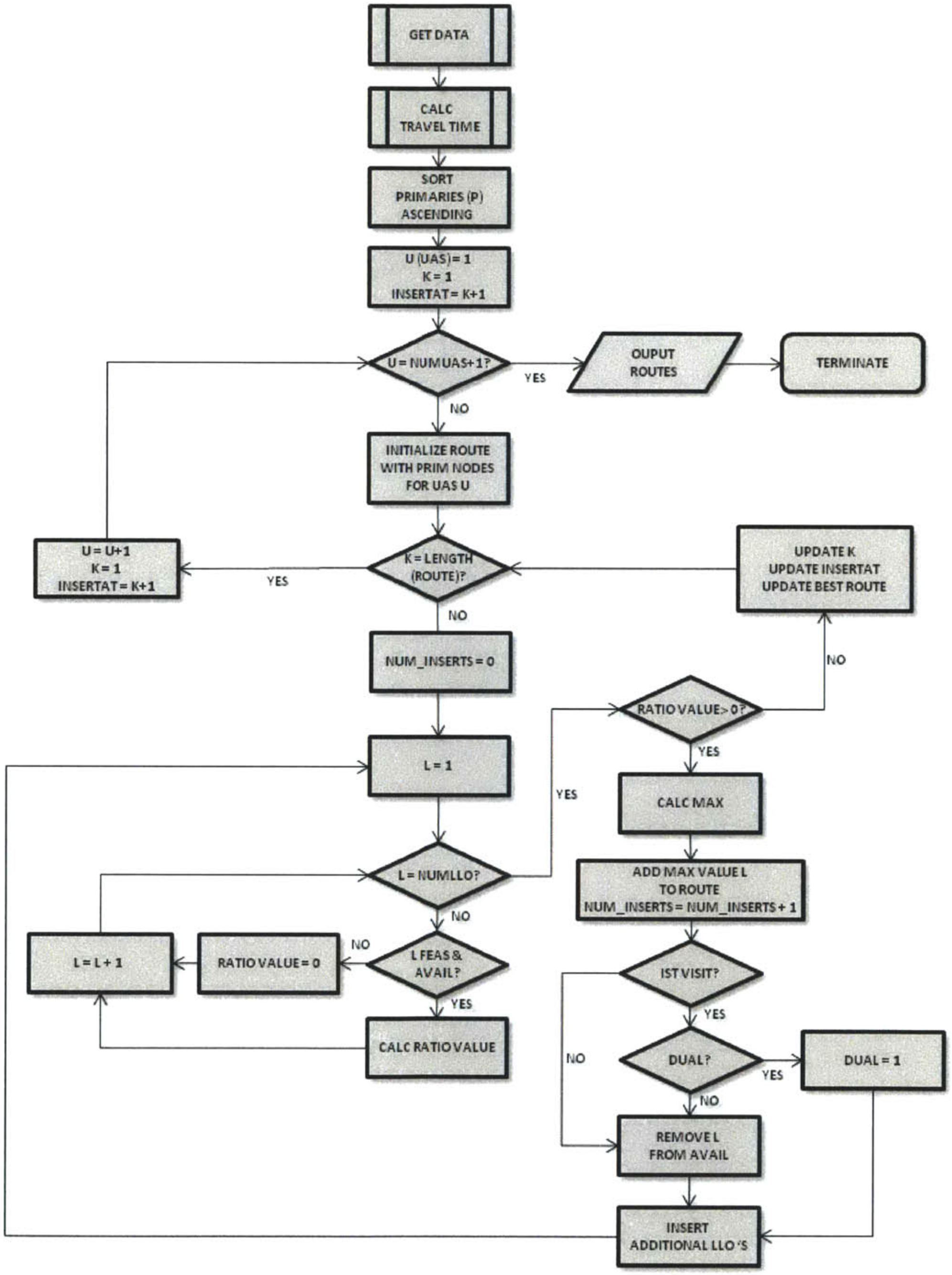
[35] US Government Counterinsurgency Guide (Washington, DC: U.S. Government Printing Office, January 2009), 11-16.

[36] Vansteenwegen, P., Souriau, W., Vanden Berghe, G., and Van Oudheusden, D. "Iterated local search for the team orienteering problem with time windows," *Comput. Oper. Res.* 36 (12) (2009) 3281–3290.

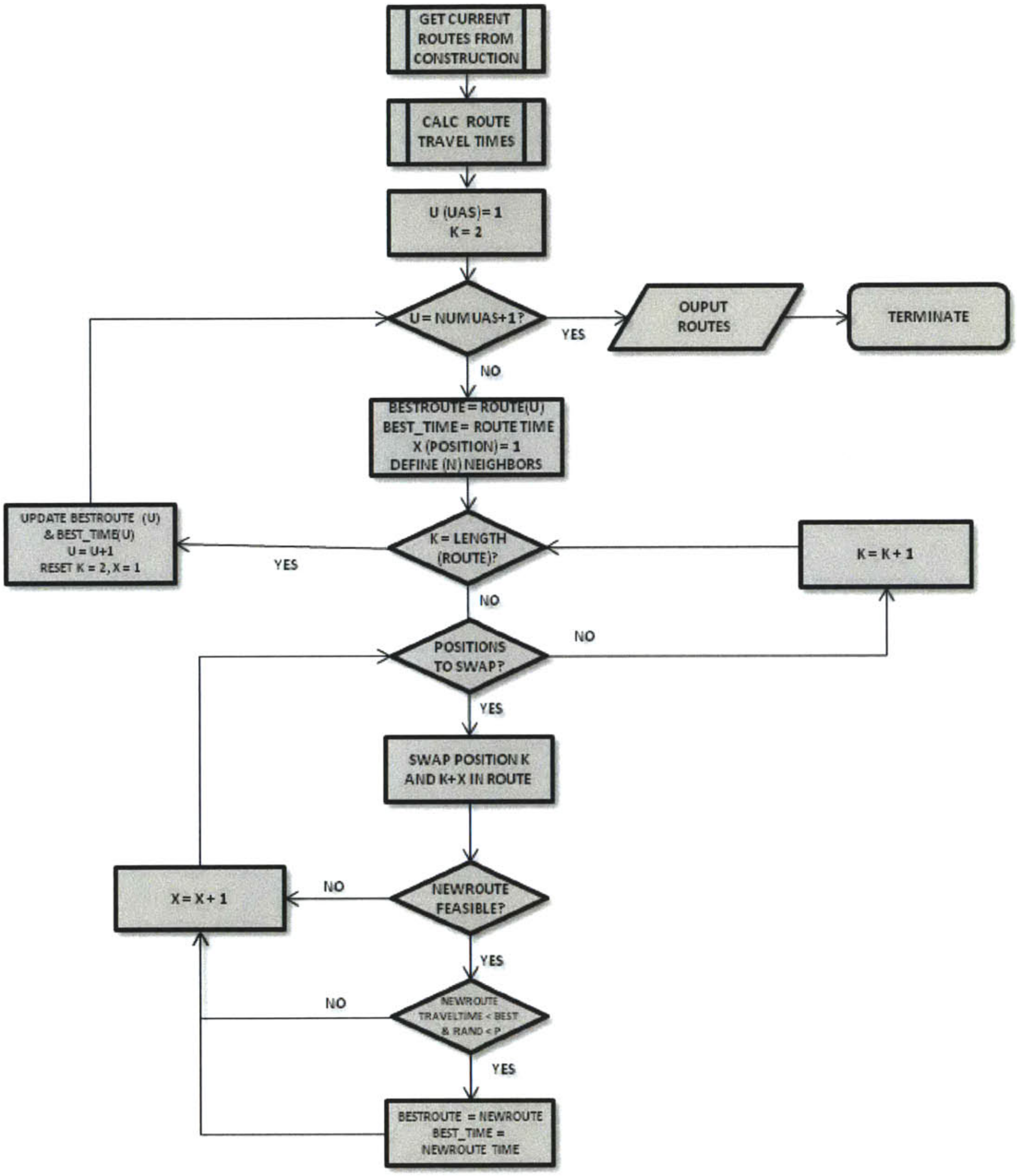
[37] Vansteenwegen, P., Souriau, W., Vanden Berghe, G., and Van Oudheusden, D. "Metaheuristics in the Service Industry," Lecture Notes in Economics and Mathematical Systems. Springer Verlag 2009.

Appendices

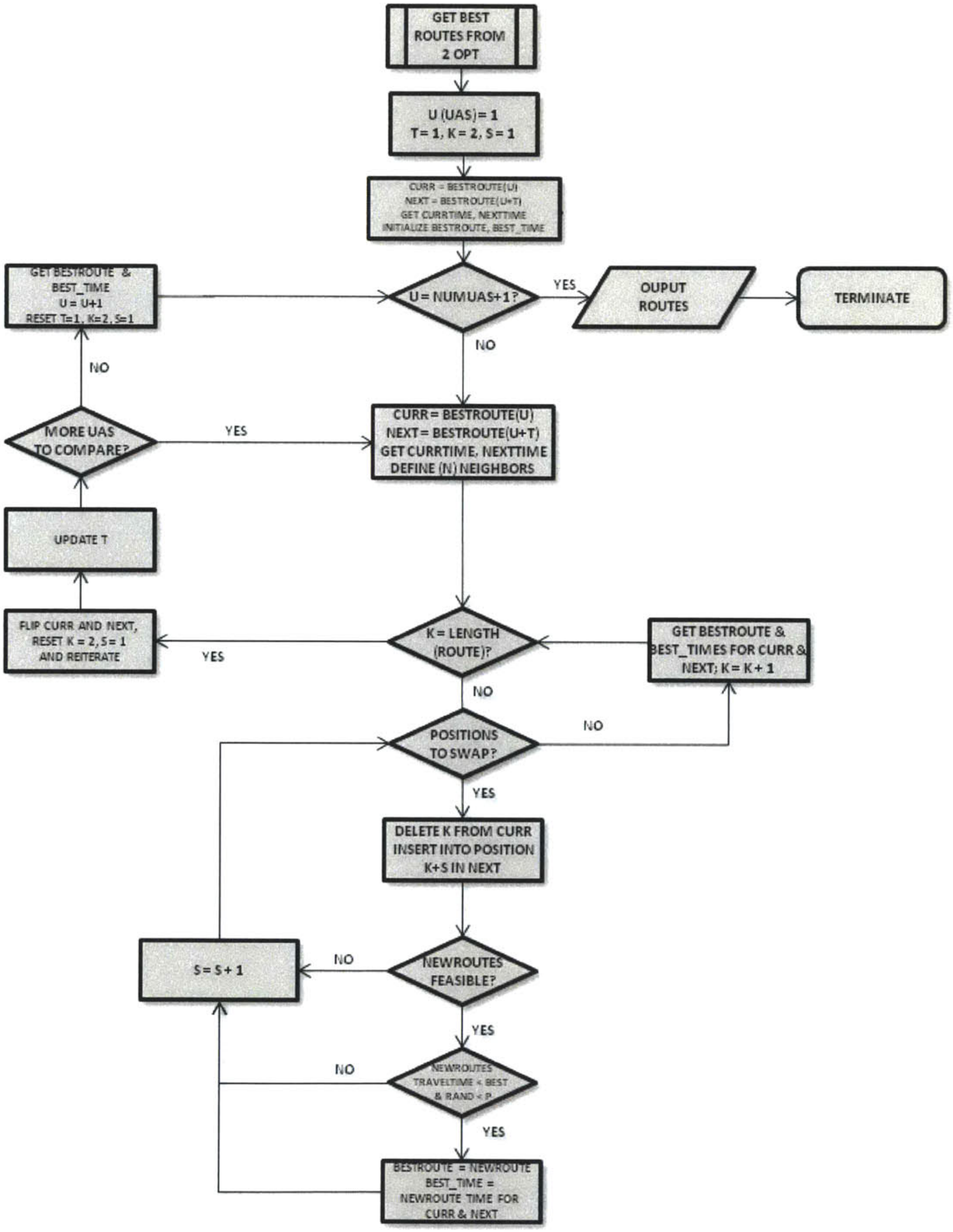
Appendix A: Insertion Based Construction Heuristic



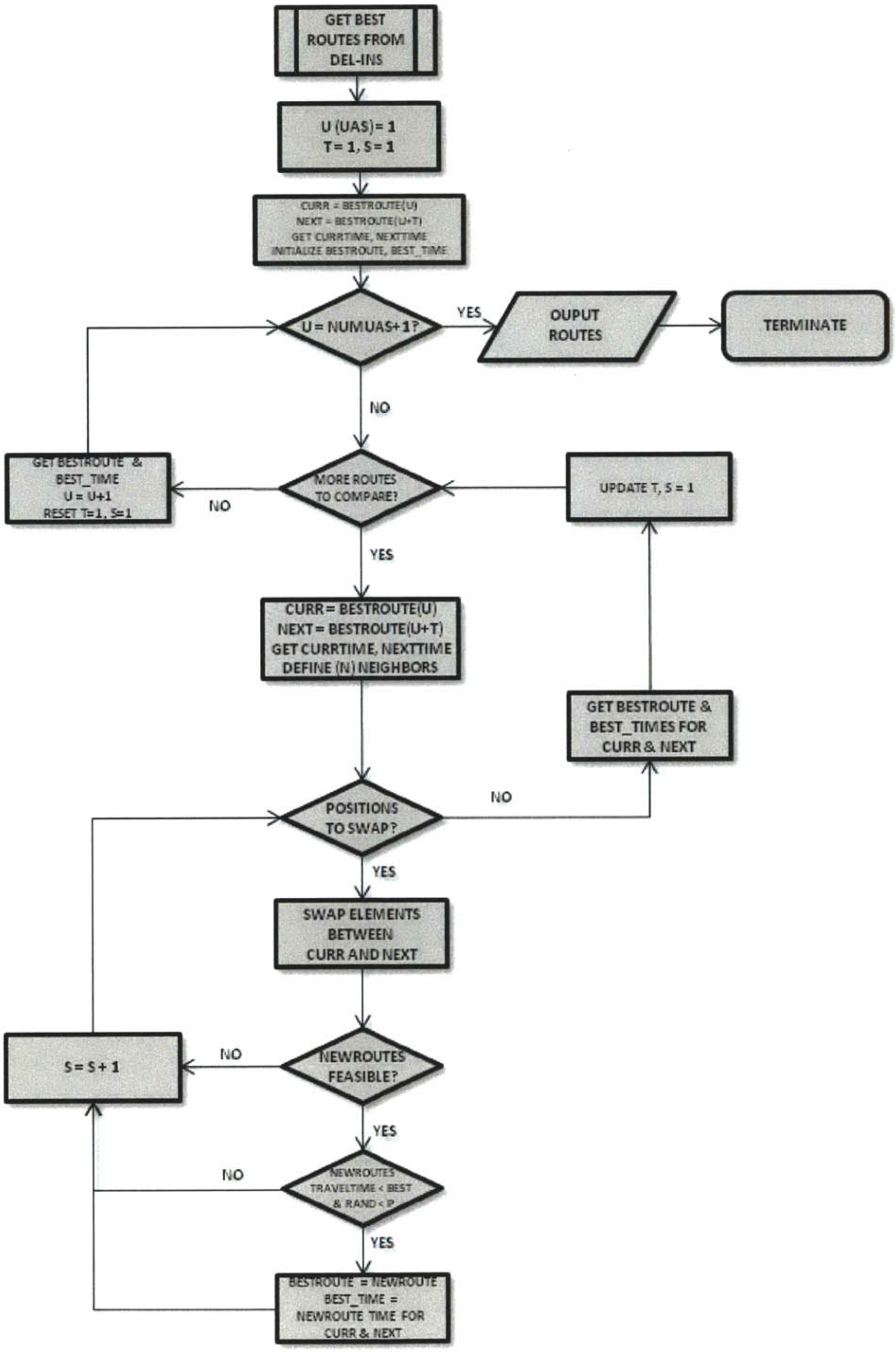
Appendix B: Intra-Route 2 Opt Improvement Heuristic



Appendix C: Inter-Route Deletion-Insertion Improvement Heuristic

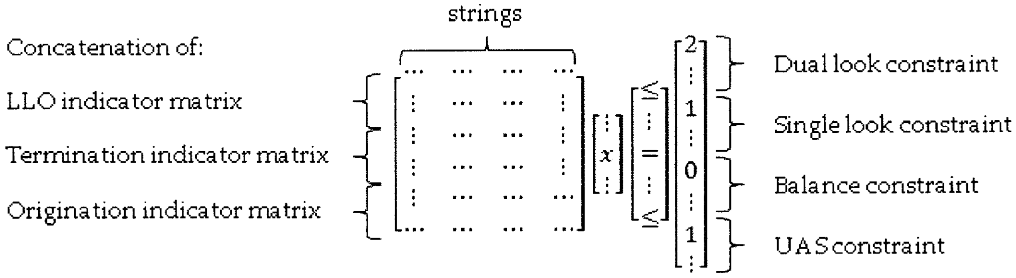


Appendix D: Inter-Route 2 Exchange Improvement Heuristic



Appendix E: Ax=b Matrix Notation for String Concept

$Ax = b$ matrix notation for String concept binary integer program



Appendix F: Results of Statistical Measures on Random Data

Uniformly Distributed Random Data Set:

Mean Distance to Convex Hull Vertices (km)				
Tests	UAS 1	UAS 2	UAS 3	UAS 4
4U5N	2.75	1.6	2.75	1.05
4U10N	4.20	2.44	2.40	2.43
4U15N	2.20	2.69	2.69	2.47
4U20N	3.37	2.83	2.52	2.94
4U25N	3.04	1.95	1.76	2.26

Tests	**Principal Component Analysis**		
4U5N	1.28	0.88	0.82
	0.429	0.725	1
4U10N	1.190	1.120	0.680
	0.397	0.771	1.000
4U15N	1.180	1.060	0.740
	0.395	0.750	1.000
4U20N	1.190	1.010	0.780
	0.390	0.730	1.000
4U25N	1.300	0.980	0.710
	0.435	0.762	1.000

Border Scenario Random Data Set:

Mean Distance to Convex Hull Vertices (m)				
Tests	UAS 1	UAS 2	UAS 3	UAS 4
4U5N	1.75	1.5	1.4	1.2
4U10N	6.45	4.16	3.12	3.06
4U15N	4.36	2.72	2.51	3.28
4U20N	5.07	3.75	3.20	4.24
4U25N	3.46	3.10	2.97	3.18

Tests	**Principal Component Analysis**		
4U5N	1.76	1.07	0.15
	0.589	0.9485	1
4U10N	1.770	0.950	0.270
	0.590	0.909	1.000
4U15N	1.440	0.980	0.560
	0.481	0.811	1.000
4U20N	1.270	1.010	0.700
	0.424	0.763	1.000
4U25N	1.370	0.920	0.700
	0.458	0.766	1.000