Coordinated Dynamic Planning for Air and Space Operations

by

Matthew Christian Wroten

B.S. Operations Research, Mathematics
United States Air Force Academy, 2003

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Signature of Author: ______________________________
Sloan School of Management
Interdepartmental Program in Operations Research
06 May 2005

Approved by: ______________________________
Stephan E. Kolitz
Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by: ______________________________
John-Paul Clarke
Associate Professor, Department of Aeronautics and Astronautics
Thesis Advisor

Accepted by: ______________________________
John N. Tsitsiklis
Professor, Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

[This Page Intentionally Left Blank]

Coordinated Dynamic Planning for Air and Space Operations

by

Matthew C. Wroten

Submitted to the Sloan School of Management
on 06 May 2005 in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Operations Research

ABSTRACT

Planners of military air and space operations in a battlefield environment seek to allocate resources against targets in a way that best achieves the objectives of the commander. In future conflicts, the presence of new types of assets, such as tactical space-based sensors and Operationally Responsive Spacelift (ORS) assets, will add complexity to air and space operations decisions. In order to best achieve objectives, planners of different types of assets will likely need to work collaboratively when formulating tasking for their resources. The purpose of this research is to investigate the challenges of air and space collaboration and to quantify its potential benefit.

We model a future threat scenario involving a rogue nation with Weapons of Mass Destruction (WMD) capability and a significant air defense force. We consider three separately-controlled resource groups – aircraft, satellites, and ORS assets – to combat the target threat. In addition, we formulate a top-level coordination controller, whose job it is to effect collaborative decision-making among resource groups.

Using a combination of pre-existing software and new algorithms, we develop the Coordinated Dynamic Air and Space Operations Control System (*CDASOCS*), which simulates controller-generated plans in a battlefield environment recurring over multiple planning periods. New algorithms are presented for both the top-level coordination controller and the ORS controller. The benefits of resource coordination in *CDASOCS* are demonstrated in three main experiments along with several parameter variation tests.

Thesis Supervisor: Professor John-Paul Clarke
Title: Professor of Aeronautics
Professor, Department of Aeronautics and Astronautics

Technical Supervisor: Dr. Stephan E. Kolitz
Title: Principle Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.

[This Page Intentionally Left Blank]

ACKNOWLEDGEMENTS

I would like to express my appreciation to all of the individuals that have helped me over the past two years. First, I want to thank Dr. Stephan Kolitz of Draper Laboratory. The deep commitment he has to his students transcends the expectations of an advisor. His wisdom and guidance have helped me grow as a student and as an individual.

I would also like to thank Professor John-Paul Clarke for his willingness to ask tough questions and challenge assumptions. He has a remarkable ability to provide a unique perspective to any problem and I thank him for being my advisor.

Next, I want to thank Dr. Charlie Strauss, Dr. Owen Deutsch, and Dr. Pete Scheidler of Draper Laboratory for their help on the GC2IAO project. They were always eager to answer my questions, patient in teaching me new concepts, and gracious with their time.

I would like to thank MIT, The Charles Stark Draper Laboratory, the Air Force and the Air Force Academy for giving me this chance to further my education. Also, I thank Lt. Col. Andrew Armacost for being an inspiring teacher and for encouraging this opportunity.

I want to thank my friends at Draper, especially Caleb and Randal, for the good conversations over coffee, the time spent together getting through the rigors of classes, and the exceptional companionship – Y'all are gonna be missed!

Next, I want to thank my family for their abiding love and support. Mom and Dad, Michael and Kathryn, you have always been very encouraging of whatever I do; I have truly been blessed with a great family.

Lastly and most importantly, I thank my wife, Cori. I could not have done this without you. Your love and devotion mean more to me than anything in the world.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Internal Company Sponsored Research Project 12596-001, Global Command and Control for Integrated Air Operations (GC2IAO).

06 May 2005

Matthew C. Wroten, 2nd LT., USAF — Date

[This Page Intentionally Left Blank]

Table of Contents

[This Page Intentionally Left Blank]

List of Figures

List of Tables

[This Page Intentionally Left Blank]

1 Introduction

In a military campaign, air and space operations planners have a wealth of information to consider when producing operational plans. Commander's objectives, target lists, intelligence reports, and analyses of capabilities are just some of the factors that contribute to air and space decision-making. Using these inputs, it is the goal of planners to assign resources to missions in the way that best achieves the commander's objectives with as little risk and cost as possible. In the case of air operations, these resource assignment decisions are guided by a process known as the Targeting Cycle.

1.1 The Targeting Cycle

Joint Publication 3-60, *Joint Doctrine For Targeting,* defines a target as "an area, complex, installation, force, equipment, capability, function, or behavior identified for possible action to support the commander's objectives, guidance, and intent" [14]. There are two major categories of targets within military operations: planned and immediate. Planned targets are those entities that have been labeled as targets since before the execution phase of the current battle plan. Immediate targets are those entities that become targets during the execution phase. If a target poses an imminent danger, is rapidly fleeting, or is labeled as high-value, it can be further classified as a Time-Sensitive Target (TST). In Figure 1-1, the different target categories are displayed. Note that TSTs can come from any target category.

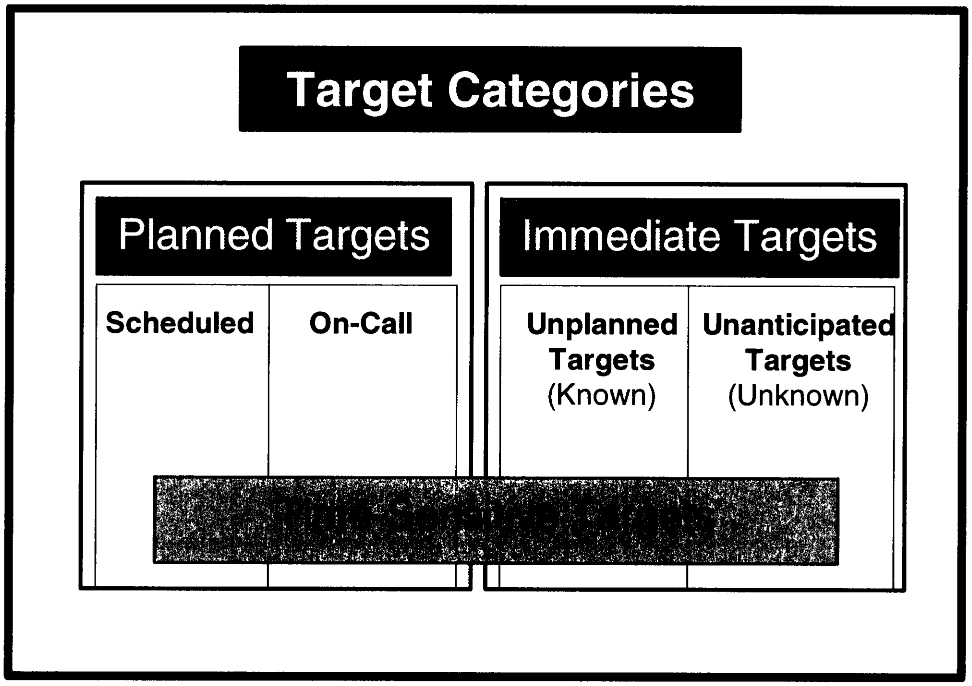

Figure 1-1: Target Categories (adapted from [14])

Operations centers create missions for planned targets using the Joint Targeting Cycle. This cycle outlines to commanders the specific steps involved with targeting. It is a model that is used across a range of military domains and so is discussed below without any specific context. Phases of the Joint Targeting Cycle are shown in Figure 1-2.

In the first phase of the cycle, *Commander's Objectives, Guidance, and Intent,* it is the job of the commander to present "clear, quantifiable, and achievable objectives [that] lead to the successful realization of national security goals through a targeting solution" [14]. An important component of this phase is Identifying Centers of Gravity (COG). Centers of Gravity, as defined by Joint Pub 3-0, are "those characteristics, capabilities, or sources of power from which a military force derives its freedom of action, physical strength, or will to fight." They might include actual military targets or less tangible forces such as public support. In addition to identifying COGs, the commander must provide objectives as well as Measures of Effectiveness (MOE) for each COG. MOEs are metrics used to quantify achievement of the objectives.

In Phase 2, *Target Development, Validation, Nomination, and Prioritization,* planners identify targets, validate them as lawful and viable targets, nominate them to higher channels for approval, and prioritize them according to importance. The result of this phase is a prioritized Target Nomination List (TNL) that includes the desired outcome for each target along with any stipulations.

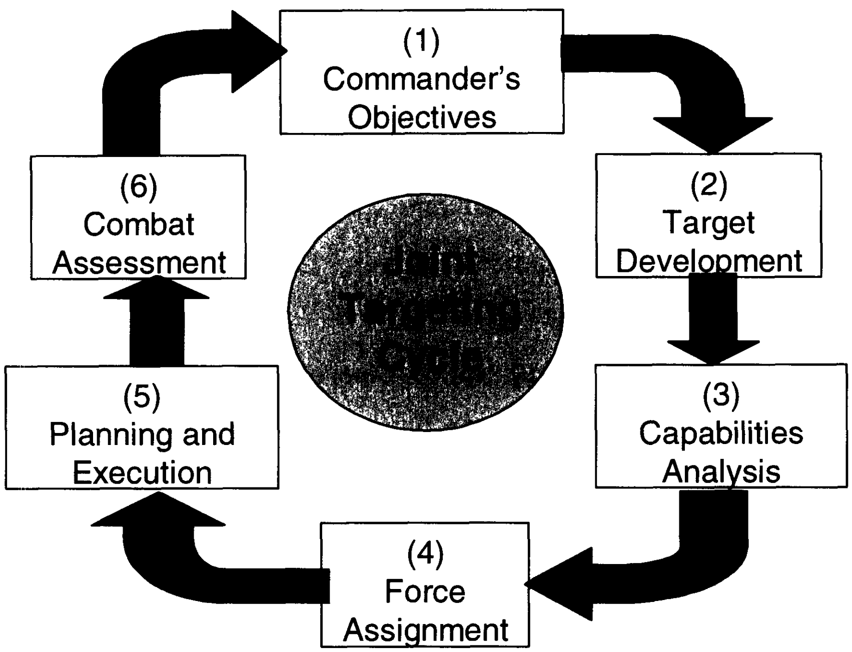

Figure 1-2: Joint Targeting Cycle (adapted from [14])

Planners match specific forces to use against each target in phase 3, the *Capabilities Analysis* phase. Possible mission combinations are analyzed using mathematical models that take into account weapon system ability and target characteristics. Once this is accomplished, planners form a recommendation of the most effective grouping of resources to use against each target.

In Phase 4, *Commander's Decision and Force Assignment*, the commander reviews and finalizes target recommendations to ensure a synergistic integration of resources.

In Phase 5, *Mission Planning and Force Execution,* mission specific details such as airplane routing and rendezvous points are determined by planners. Once this is completed, the final plan, as formulated in phases one through five, commences.

Intelligence experts consolidate data from various sources and evaluate the success of missions in Phase 6, the *Combat Assessment* phase. This phase is composed of three sub-parts: Battle Damage Assessment (BDA), Munitions Effectiveness Assessment (MEA), and future targeting recommendations [14].

1.2 Thesis Overview

This scope of this research involves air and space operations that are guided by the Targeting Cycle. The thesis is organized into six chapters. The theme of each chapter subsequent to this one is as follows:

Chapter 2 – Joint Air and Space Operations: We present a future threat scenario and further specify the Targeting Cycle under the operational framework that would likely deal with the threat. We outline a Concept of Operations for the scenario, including descriptions of the types of resources assumed and a specific command and control structure. Finally, we define the Air and Space Operations Planning Problem *(ASOPP)* to which the rest of the thesis is directed.

Chapter 3 – Model Development: A functional model of the problem is presented to frame it from a mathematical perspective. Modeling assumptions are made and the inputs, decisions, objectives, and constraints of every functional component are discussed. Two of the functional components given in this chapter represent new contributions toward researching the coordination of air and space resources.

Chapter 4 – Algorithms and Formulations: The purpose of this chapter is to formulate new algorithms that provide the decision-making logic to the functional components of Chapter 3. We research previous formulations and solution methods that have been, or could be applied to, the Targeting Cycle. We list certain restrictions on the algorithms arising from software constraints. Also, three pre-existing pieces of software that are employed in the system are described. Lastly, we present the five algorithms/formulations that represent new contributions in this thesis.

Chapter 5 – Testing and Analysis: In this chapter, the algorithms of Chapter 4 are tested on a simulation, and their results analyzed. First, a complete picture of the software implementation and integration is shown. Then, a description is provided of the parameter values used in various testing scenarios. Next, three main experiments are used to quantify the benefit of adding space to air operations as well as coordinating among resources. Finally, a series of miscellaneous tests are conducted to analyze the effects of varying individual parameters within each component of the system.

Chapter 6 – Conclusions and Future Work: In the last chapter, a summary of what is accomplished is given, including major insights gained from the research. A final section is devoted to areas of future research for possible extensions to what is done here.

[This Page Intentionally Left Blank]

2 Joint Air and Space Operations

This thesis focuses on air and space operations that are planned for and directed by a Joint Air Operations Center (JAOC) in the year 2025, where a JAOC is the assumed entity through which all resource tasking is generated. In this chapter, a war-time scenario is presented in which the JAOC, following doctrine from the targeting cycle, is tasked to subdue an enemy threat using military force. We describe assumptions regarding the general threat scenario and available resources. We present a Concept of Operations (CONOPS), which is a broad outline of the operation from a military perspective, and discuss the organization of the JAOC. Because some resources in the scenario do not yet exist, the CONOPS is necessarily hypothetical in nature. It is the goal of this chapter to specify the threat scenario and CONOPS and to define the operational problem to which the rest of the thesis is directed.

2.1 Scenario

Within the next 30 years, it is possible that the US military will be called upon to suppress a hostile nation seeking to proliferate and/or detonate Weapons of Mass

Destruction (WMD). Such a nation might have a large conventional military, significant Integrated Air Defense Systems (IADS), and could pose an immediate threat to the US, or to a US ally. If this situation arises without warning, the US might not have any significant land forces staged in the region. Given a National Command Authority directive to completely negate the threat through the use of military force (with certain economic, political, and environmental stipulations), it would then become the job of the Joint Force Commander (JFC) in charge of all forces in that region to formulate military objectives, decide which forces to utilize, and carryout the execution of the ensuing military campaign. Because of the urgency of the situation, it is likely the JFC would rely heavily on resources that are quickly accessible, namely air and space power, until other resources arrive. Under these assumptions, the following military objectives are plausible for the first few days of the campaign:

1. **Search for and negate high value targets (WMD sites, Command and Control facilities, etc.)**
2. **Search for and negate other key conventional targets (artillery, infantry units)**
3. **Establish Air Superiority**
4. **Avoid civilian casualties and damage to the economic infrastructure**
5. **Prepare for a subsequent US/coalition conventional force invasion**

In addition to tasking the JAOC with these objectives, the JFC would also outline, for each objective, quantifiable measures of success. The remainder of this chapter is used to describe how the JAOC would likely function in this setting. It is important to understand the operation of, and decision making in a JAOC as well as information sharing that goes on between air and space planners. We describe the JAOC's available resources, its command and control structure, and aspects of its day-to-day operation.

2.2 Resources

New types of air and space resources will likely be developed and made operational within the next 20 to 30 years. Rapidly advancing technology and the need to respond to a variety of different threats will push this development and, as a result, will cause some of the current inventory to become obsolete. In an attempt to remain as up-to-date as possible, we include in this research some vehicles that are presently operational as well as some still in the concept or design phase. In Figure 2-1, the three categories of resources of this future operational scenario are shown. For each category, a section is devoted to explain the assumed capabilities of that resource set. Resources include:

1. **Space Force Enhancement assets (ISR only)**
2. **Air assets (ISR and Strike)**
3. **Space Force Application weapon systems (Strike and one-time ISR)**

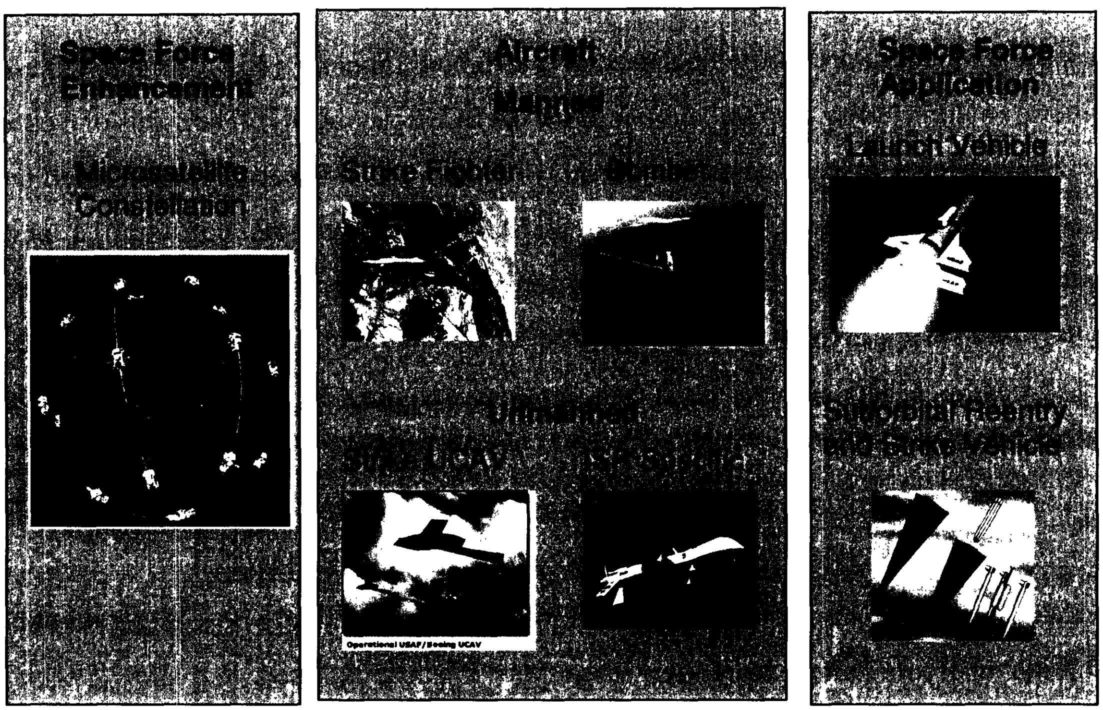

Figure 2-1: Resources of a future JAOC

The range of available resources is confined to platforms capable of performing strike and/or Intelligence Surveillance and Reconnaissance (ISR) missions. A strike mission involves the planned destruction of one or more targets through the use of air-to-ground (or space-to-ground) munitions. An ISR mission, on the other hand, seeks to gain information about a target or an area of interest, usually through the use of imaging sensors. ISR missions either involve the continuous, or near continuous, collection of data over a period of time (surveillance) or the one time tasking of a resource to image a target and return to base (reconnaissance). It is possible for a single resource to perform both strike and ISR on the same mission. Finally, it is assumed that an unlimited number of mission-support vehicles, e.g., air refueling aircraft, exist and are always available when needed.

2.2.1 Space Force Enhancement Assets

Space Force Enhancement, as laid out in Joint Publication 3-14, *Joint Doctrine for Space Operations*, includes operations that "multiply joint force effectiveness by enhancing battlespace awareness and providing needed warfighter support" [12]. Currently, these operations are directed by a space version of the JAOC (Space AOC). The types of Space Force Enhancement assets considered in this research are ISR-capable microsatellites (microsats) that are placed in a defined constellation. Microsats constitute a new generation of imaging satellites that are currently still being developed. According to [16], they will be approximately three feet in diameter, 1.5 feet tall, and weigh no more than 220 pounds. There are two main benefits to developing and using microsats over existing conventional satellites. First, the cost to launch a microsat is much less due to its decreased weight. Secondly, the fuel requirement, once on orbit, is much reduced. ISR microsats are being designed to launch into Low-Earth Orbit (LEO) [23].

Because of the low-cost nature of microsats, there will likely be a larger number of satellites in future ISR constellations than what is currently used. We allow for two possible constellations in this scenario – one with 36 satellites and one with 72. Also, the following four assumptions are made regarding satellite orbits: 1) all orbits are circular and have the same altitude, 2) the angular spacing of satellites within an orbital plane is constant, 3) the relative phasing between planes is constant, and 4) once in orbit, satellite maneuvering is confined to station-keeping. Under these assumptions, there will be some trade-off between target coverage, the number of satellites in orbit, and the quality of sensor measurements [1]. Achieving effective target coverage requires either more satellites with a lower altitude or fewer satellites with a higher altitude. While the former case increases cost, the later decreases the quality of sensor measurements. At any rate, this is a tradeoff that satellite planners make *well before* the start of the campaign.

It is assumed that each satellite carries exactly one sensor. The three types of sensors most likely to be put on microsats include: infrared (IR), visual, and radar. IR sensors record invisible infrared light and are used primarily to detect medium to long-range missile launches, not for ISR purposes. Visual sensor technology can provide reasonably high-resolution images (if the satellites are close enough to the earth) but is limited by both weather and daylight. Radar sensors can also provide high-quality images and are not affected by clouds or darkness. However, the more powerful the radar sensor, the more fuel it consumes. For ISR microsatellites, there seems to be a tradeoff between the relatively low fuel consumption of visual sensors versus the ability to operate in any weather conditions/time of day of radar sensors. The only assumption made here regarding each satellite's sensor is that it has high enough resolution to positively identify, with reasonable accuracy, the types of targets considered in this scenario.

2.2.2 Air Assets

Air assets capable of ISR and strike include attack, bomber, fighter, and UAV aircraft. Table 2-1, while not an exhaustive inventory, lists operational and developmental aircraft that serve in this capacity. Only aircraft that can directly further the joint campaign objectives are included. Other aircraft used for component-specific roles, such as the S-3B Viking and the A/OA-10, whose primary mission is Close Air Support to troops on the ground, are excluded. From a modeling perspective, each aircraft has characteristics that distinguish it from the rest, (e.g., cruising speed, maximum distance before refueling, vulnerability, munitions and payload capabilities, etc.).

Although most of the needed aircraft might not initially be available to the campaign, they are capable of reaching the theater quickly. The Military Analysis Network, as part of the Federation of American Scientists, claims that the Air Force has the capability to deploy seven to eight fighter-wing-equivalents (FWEs), composed of 72 aircraft each, to a distant region in a matter of a few days [3]. Moreover, both the Air Force and Navy follow the concept of forward deployment and have some air forces deployed around the world at any given time. Thus, while it is likely there will be some forces already in the region, the rate at which new aircraft become available to the campaign is dependent upon the campaign's geographic location and the positioning of US forces.

	ISR and Strike Aircraft				
	[redacted]			[redacted]	
	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
USAF	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
USN	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

Table 2-1: Partial Listing of Current and Planned ISR and Strike Aircraft

2.2.3 Space Force Application Assets

The Space Force Application mission is defined by Joint Publication 3-14 as "operations [consisting] of attacks against terrestrial-based targets carried out by military weapons systems operating in or through space" [12]. Current research and development is being done on the concept of Operationally Responsive Spacelift (ORS). The ORS system would involve launching a vehicle into orbit and having it ready to execute its mission within hours of being tasked. According to [9], the Air Force could do a test-launch of the ORS system as early as 2014. The Space Operations Vehicle (SOV) is one candidate for the launching vehicle. The SOV, as defined in [22], is designed to be a low-earth orbit launch and delivery system. Different sources cite the launching payload at anywhere from 10,000 to 40,000 pounds [23], [22]. The SOV is able to carry a variety of different payloads, some in combination. We consider two possible payload types – the Common

Aero Vehicle (CAV), and a microsatellite of the same type as those in the existing satellite constellation.

One reason why the SOV-CAV/microsat combination might be developed over other types of intercontinental missiles and launch vehicles is that the CAV can be recalled before striking a target. This essentially means that a CAV could be "launched toward a potential target even before a final decision was made to attack" [20]. Also, if launched in combination with a satellite, the payload still in space can be used to conduct other surveillance [21]. In addition, there are political considerations in favor of the CAV over a conventional version of the ICBM. Some countries might respond strongly if the military were to continue development of the same launch vehicle that was once used for nuclear warheads. While a CAV could be used for a strike mission, an ORS launch of a microsat could aid in providing quick reconnaissance on a specific target and then join the existing satellites to add to the overall capability of the constellation.

According to [17], the conceptual CAV weighs less than 2000 pounds, can carry 800-1000 pounds of weapons, and can be launched into either a suborbital or orbital trajectory. If launched into orbit, it remains there until tasked with striking a target, at which point it re-enters the atmosphere and releases its weapons against the target. Payload options include smart bombs, deep penetrators, or precision anti-armor weapons. The CAV is capable of striking moving targets as well as hardened or deeply buried targets. The predicted range of the weapon, once released from the SOV is listed at 3000 nautical miles. The cross-range is listed at 2000-3000 nautical miles [19]. The CAV provides a relatively cheap and accurate option to a commander who needs to strike a target quickly and where sending aircraft missions is infeasible. The author of [9] suggests that a CAV could launch and strike anywhere in the world within two hours.

In summary, for use at the discretion of the JAOC, we have assumed aircraft capable of ISR and strike, microsatellites for ISR, and an ORS system for launching both ISR microsatellites and suborbital strike vehicles. Next, we describe the command structure used to task, monitor, and re-task these resources.

2.3 Command and Control

In modern military operations and under our assumptions, a JFC exercises complete Operational Control (OPCON) over all resources in the region. OPCON, as defined by [11], is "the authority to perform those functions of command over subordinate forces involving organizing and employing commands and forces, assigning tasks, designating objectives, and giving authoritative direction necessary to accomplish the mission." One functional area for which it is not clear who will have the OPCON in any given scenario is space forces. The control of space forces in a military operation either resides with the JFC or with US Strategic Command (of which the former US Space Command is a part) [12]. It is assumed for this research that the OPCON of all resources in the scenario belongs to the JFC. This is reasonable given the nature of the threat and types of resources being requested.

While the JFC exercises OPCON over all assets, lower level commanders may have tactical control over subsets of the assets. Tactical control is defined as the authority over day-to-day planning and tasking of resources. Currently, the Joint Forces Air Component Commander (JFACC) exercises tactical control over most air assets in a campaign. The JFACC is responsible for planning aircraft missions as well as determining which targets to hit and when to hit them [13]. However, assets used for service-specific missions such as anti-submarine reconnaissance are not under the JFACC's control. The JFACC is responsible for directing the air campaign, consistent with the objectives of his or her commander, the JFC. Still, the question remains as to who has tactical control over space operations

At present, no doctrinal space component exists in joint force operations. Ricky Kelly, in his study entitled *Centralized Control of Space: The Use of Space Forces by a Joint Forces Commander,* recommends space forces in the future be doctrinally placed under the control of the JFACC [15]. The current directive from Joint Publication 3-14, is that tactical control of space forces will either remain at the top level (JFC) or be placed under a component commander [12]. For the remainder of this research, it is assumed the tactical (day-to-day) control of all resources, including satellites, rests with the air component commander – the JFACC.

2.3.1 Day-to-Day Planning

The JFACC must develop a Joint Air Operations Plan (JAOP) at the start of a campaign that outlines a strategy for integrating air operations into the larger operational picture. While the JAOP is completed before the beginning of a campaign and involves broad operational goals, the day-to-day operations such as mission planning, execution, and analysis are governed by a model known as the Joint Air Tasking Cycle. We add space resources to this model and label it the Joint Air *and Space* Tasking Cycle (JASTC). Phases of the JASTC are shown in Figure 2-2. Solid arrows represent the output from phases (which are inputs into subsequent phases) and hashed arrows represent the coordination/guidance necessary to complete the phase. Coordination and guidance come from either the JFACC or the intelligence analysts that are in charge of the Intelligence Preparation of the Battlespace (IPB). The IPB is an estimate about the state of every entity and area on the battlefield at the current time. Information flows of the JASTC are listed in Figure 2-3.

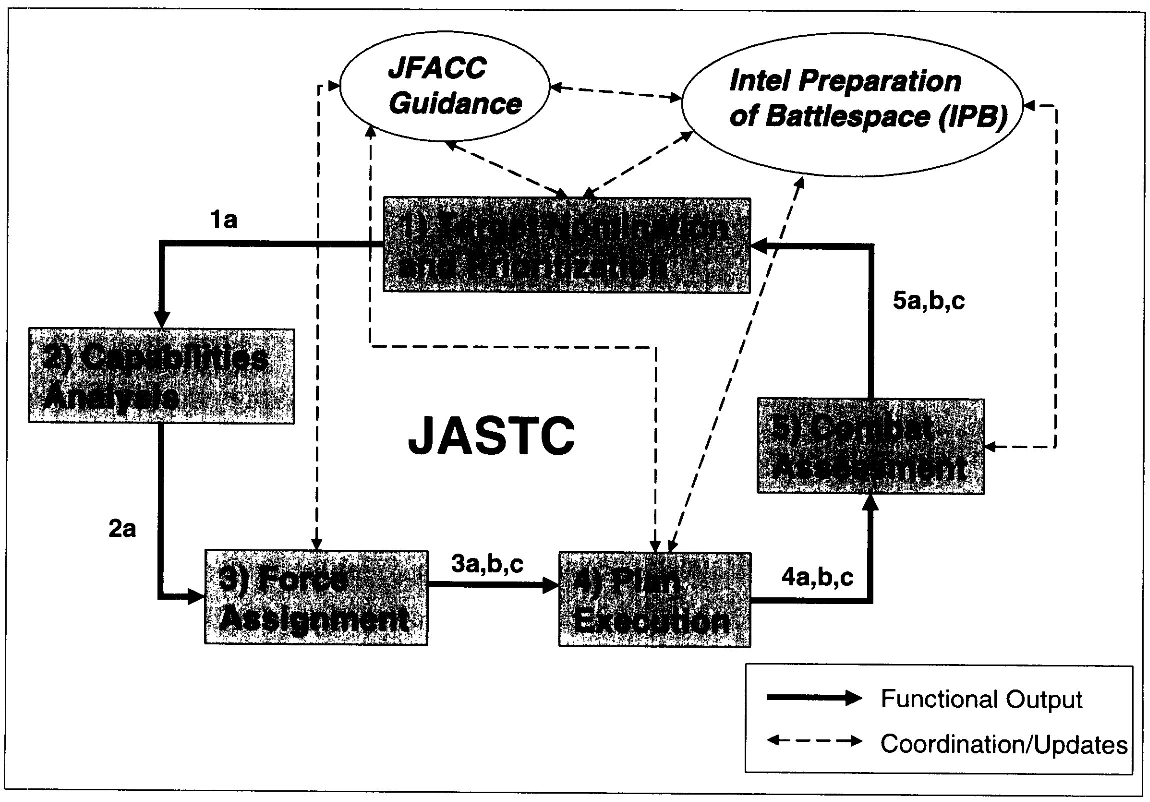

Figure 2-2: Phases of the Joint Air and Space Tasking Cycle (adapted from [13])

Outputs from Phases

1) Target Nomination and Prioritization
   a) Joint Integrated Prioritized Target List
2) Capabilities Analysis
   a) Weapon and Target Pairing
3) Force Assignment
   a) Routing and Timing decisions
   b) Resource allocation decisions
   c) Weaponeering decisions
4) Plan Execution
   a) In-flight reports
   b) Discovery of Time Critical Targets (TCTs)
   c) Initial Bomb Damage Assessment (BDA)
5) Combat Assessment
   a) BDA
   b) Munitions Effectiveness Assessment (MEA)
   c) Future Targeting Recommendations

IPB and JFACC Information

Intel Preparation of Battlespace (IPB)
1) Asset Data (num, location, state)
2) Target Data (num, location, state)
3) Weather, Political/Cultural Data
4) Adversary Courses of Action

JFACC Guidance
1) Importance of Geography
2) Importance of Time
3) Risk Tolerance
4) Planned Time Critical Targets (TCTs)
5) End State

Figure 2-3: Information Flows in the JASTC

The JASTC is a direct application of the Joint Targeting Cycle. The information flows are not discussed in detail here, although seeing a visual representation of where they fit into the JASTC is useful as they are fundamental to the functional model of Chapter 3.

The main emphasis of the JASTC is the production of Air Tasking Orders (ATOs). ATOs are generated on a recurring, 72-hour basis and fit into phases 2), 3) and 4) of the JASTC. ATOs encompass all tasking of resources to missions involving *planned* targets. The evolution of an ATO from start to finish involves 48 hours of planning and 24 hours of execution. Also, a new ATO is started every 24 hours. Thus, at any given time (after reaching "steady state"), there are two ATOs in the planning phase and one in the execution phase, as seen in Figure 2-4.

Although the ATO process is currently on a 72-hour schedule, this could decrease significantly in the next 20-30 years. As ISR assets become more proficient and information and data gathering capability increases, it makes sense for the ATO to shift to a shorter cycle. This gives the commander a chance to update and change the objectives for different campaign phases throughout the day. It also allows more updated versions of IPB to be factored into the plan. For these reasons we hypothesize that future ATOs will be on cycles in the range of 30 minutes to two hours.

An ATO requires numerous decisions to be made about each mission including, but not limited to, the following:

1. **What actions to take against each target**
2. **What resources to use against which targets**
3. **When to start each mission and when to take the prescribed actions against each targets**
4. **What resources to use to search in Areas of Interest for new targets**
5. **What routing to use to rendezvous, get to and from the target, etc.**

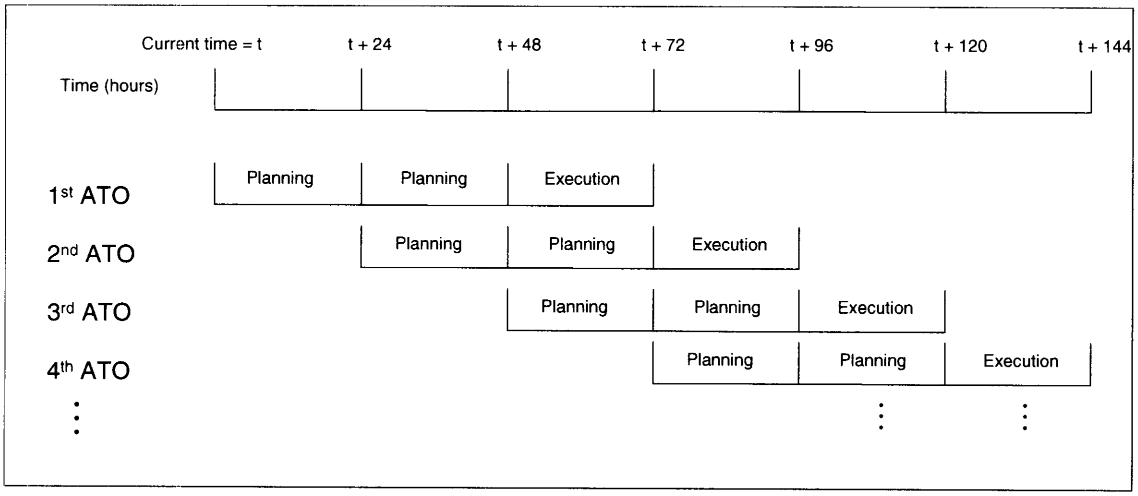

Figure 2-4: ATO cycle

It is worth noting that different divisions within the JAOC control the ATO at different times. In the planning phase, the Combat Plans division of the JAOC is in charge of ATO production. Once the ATO shifts into execution, the Combat Operations division takes over and is responsible for execution and monitoring. In this phase, new information such as in-flight reports, time-sensitive target discovery, and initial BDA might cause the re-tasking of already assigned forces to adapt to the changing environment. Also, part of producing and updating an ATO involves continually analyzing the IPB. ATO planners, using this IPB information, seek to produce the plan that will most closely follow the commander's stated guidance and objectives. In our scenario, we assume these objectives include maximizing total enemy destruction and minimizing attrition, operating cost, and time until completion.

2.3.2 Coordination between AOCs and AOC Divisions

Even though we assume the JFACC has tactical control over both air and space operations, it is possible that the tasking of air and space resources could come from different sub-organizations residing in separate physical locations. In present-day operations, there exists a Space AOC at Vandenberg AFB, CA, that is separate from any JAOC and from which space tasking would originate in an overseas campaign. Given the vast amounts of satellites the US controls along with the numerous mission functions (ISR, communications, navigation, etc.), it is reasonable to assume that future Space AOCs and JAOCs will not be merged. Thus, for the ATO process to be effective, careful attention needs to be paid to coordination between the JAOC and the Space AOC, as well as between the different divisions within AOCs.

In an interview with Lt Col Scott Henderson, a former Space Squadron Commander, details were provided on the nature of how space ISR currently supports and is coordinated with air operations [10]. Nominally, air planners submit imaging and signals intelligence requests to a central tasking board based on current or projected ATO objectives. At least in the case of non-intelligence data such as space-based missile warning or navigation data, requests are made through the Space AOC. The Space AOC then assigns priority levels to the requests depending on the nature of what is requested. ISR missions consisting of Target Identification (ID) or Battle Damage Assessment (BDA) to coincide with a future projected strike time is possible if called for early enough in advance. Currently, the highest priority targets (TSTs) can be imaged within a couple of hours of the request. Lower-priority targets might take up to a couple of days. Whether these times will be improved in the near future remains unclear. However, one thing is certain. The speed with which satellite ISR requests are fulfilled is critical to the successful coordination of air and space operations.

Since neither the microsat nor the ORS system is currently operational in the military, it is difficult to know just how or what information needs to be shared between these planners. It is likely, though, that different divisions within the JAOC will be in charge of tasking (or providing requests for) different resource types. We may assume there are separate divisions for aircraft tasking, ORS tasking, and satellite ISR requests. Each division is in charge of formulating its own part of the ATO. One way to ensure divisional plans do not overlap is to partition the target set among divisions where each division is fully in charge of one group of targets. Another way, which might lead to more effectively achieving the commander's objectives, is to coordinate the processing of all targets among resources. Coordination involves the JFACC tasking JAOC divisions in a collaborative manner, and with feedback. In a coordinated system, each division produces plans that help to further not only division goals, but the joint goals as well. One example of effective coordination could be a microsat doing the initial ID of a target, an ORS CAV striking the target, and aircraft performing the BDA, all in close succession.

There are two direct benefits from the coordination of air and space operations. These include 1) increased destruction capability gained from resource pooling, and 2) increased efficiency due to a quicker flow of information. Here, increased efficiency is defined as achieving desired results at a faster rate and at a lower cost. The measurable aspects of these benefits include 1) higher overall target destruction, 2) faster rate of target destruction, 3) fewer aircraft attritions, and 4) lower resource operating costs.

2.4 Problem Statement

The *Air and Space Operations Planning Problem (ASOPP)* is defined as the problem of assigning air and space resources in a coordinated fashion to *find, identify, strike,* and *assess the damage of* enemy targets in such a way as to maximize the total enemy destruction while minimizing attrition, operating cost, and time until completion. The goals of this research are to:

1. **Define and model the higher-level, planning aspects of the ATO**
2. **Quantify the benefit of adding space operations to the ATO**
3. **Formulate effective coordination heuristics**
4. **Quantify the potential benefit of coordination within air and space operations.**

[This Page Intentionally Left Blank]

3 Model Development

In a mathematical model there is an inherent tradeoff between fidelity and computational efficiency. Given all the different types of resources described in Chapter 2 plus the numerous decisions required for each resource at each time step, a high-fidelity model of the JASTC could become intractable for even small data instances. Instead, we focus on producing a model that can generate high-level plans quickly, while allowing sublevel planners to handle detailed decisions like aircraft routing. Modeling assumptions are listed to lay the groundwork for a mathematical model of *ASOPP*. A hierarchical decision structure is proposed along with a system-level overview of functional components. Finally, the inputs, decisions, and constraints at each level of decision-making are described in depth.

3.1 Modeling Assumptions

In order to create a problem framework under which a viable solution method can be found, certain assumptions involving the problem's mathematical structure are made. While the validity of some assumptions could be questioned, they are necessary for starting with a simple, if not completely realistic, model. Proceeding with this method will leave room for more complex versions later and may give valuable insights into the real-world system. It is also an approach to obtaining approximate solutions to an otherwise intractable model. Six assumptions are listed below with descriptions and examples as necessary. It is important to remember that these assumptions are for the purpose of simplifying the construction of a solution method, not the computer simulation.

Finite State and Discrete Attributes: *The state of every object in the system must be fully characterized by a finite number of attributes.* For instance, there are three attributes – target type, target location, and target damage status – that define the state of the target. This is the *finite state* part of the assumption. The *discrete attributes* part, on the other hand, requires the value of each attribute to belong to a specified discrete set. In other words, for any target i, and any attribute a, the set of possible values that the attribute can take on, labeled S_{ia}, must have a finite number of elements, $|S_{ia}| < \infty$. Consider a target, labeled $tgt1$, whose state is partly characterized by the target's latitude. In the real world, $tgt1's$ latitude could take on any one of an infinite number of values within the borders of the battlespace, whereas in the model, this assumption confines latitude to be in $S_{tgt1,LAT}$.

Discrete-Time Markovian System Evolution: *The state of the system is only allowed to change in discrete time increments.* Any object in the system, therefore, can only transition states at a time period, $t \in T$, where T is the set of time periods in the battle campaign. Each action that is tasked by the model will commence at one of these time periods. Moreover, state transitions must be independent of past state information; each transition at time $t+1$ can only depend on the current state, s_t, and the actions, Ψ_t taken at time t [27]. This is known as the Markovian property and is written:

$$P(s_{t+1}|s_0,\ldots,s_t,\Psi_0,\ldots,\Psi_t) = P(s_{t+1}|s_t,\Psi_t) \tag{3.1}$$

Consider the location of an enemy tank. The model, under this assumption, assumes that any future location of the tank depends only on where it is now, not on where it has been previously.

Imperfect Target State Information: *The attributes of enemy targets are characterized with probability distributions.* In a battlespace environment, uncertainty abounds as to the enemy's location, capability, and intentions, so there is much value to be gained from "good" intelligence. To capture this in a model, the idea of *imperfect target state information* is introduced. Knowledge of a target attribute for any particular time is reflected in a prior distribution. Each new piece of information is incorporated through a Bayesian update, which produces a posterior distribution (See [8] for information on Bayesian statistics). This type of uncertainty is referred to as *estimation uncertainty*. In the real world, once a target is observed by an ISR asset, more confidence is gained about its location, damage status, speed, etc. The equivalent in the model is this: An observation of a target causes the distribution mean of each target attribute to improve, reflecting better accuracy. Moreover, the variance of each distribution decreases, reflecting better precision. Finally, in addition to state estimation uncertainty there is the uncertainty that comes from the simulation – whether a mission succeeds, whether an aircraft gets shot down, etc. It should be noted that the attributes of friendly resources are assumed to be known with perfect certainty.

Target State Independence: *The attributes of each target are independent of the attributes of any other target.* For instance, the fact that target a just got destroyed should have no effect on the capability of target b. This is not a particularly realistic assumption; there would be obvious effects on the capability of certain targets if a major command and control facility was destroyed. Likewise, consider an enemy airbase runway that is listed as a target. The ability of aircraft to take off and land at this airbase should be highly dependent on the damage state of the runway. In order for this assumption not to cause unrealistic results, careful attention must be taken to model targets without too many interdependencies.

Value and Constraint Proportionality: *Any feasible action can only contribute to the value of the objective function in a manner that is proportional with the level of that action.* Let x_i be the number of times that target i is to be struck. The proportionality assumption prohibits an expression such as $3\sqrt{x_i}$ from entering the objective function. Though an expression with a square root could help capture a decreasing-marginal-returns effect from striking the same target multiple times, a proportional objective function is assumed for now. The constraints of the problem are also limited by proportionality in the same way.

Value Additivity: *The contribution to the objective function of any combination of actions is the sum of individual contributions.* If the expected value for action a is 3, and the expected value for action b is 4, then the expected value for doing both a and b must be 7. An evident result of invoking this assumption is the inability to model synergistic effects of combining resources, a common characteristic of battlefield operations.

Now that the major modeling assumptions have been listed, a functional model of *ASOPP* is presented, along with inputs and outputs at each level.

3.2 Functional System

At least two methods exist that future military leaders may use to partition decision-making within a JAOC. The first way is according to resource *function*. This means one director would control missions involving the ISR function and one director would control missions involving the strike function. The directors would coordinate when functions interact. The second way, which is the method used in this thesis, is according to resource *category*. In essence, category partitioning means separate directors would control satellites, aircraft, and ORS assets, no matter the function of different resources. Although it is not clear which method will prevail in the future, one attractive feature of category partitioning is that resources never overlap; the control of a single resource on a mission resides with one director. Alternatively, in a function partitioning, the control of a single resource is split if there are aircraft that perform both ISR and strike.

3.2.1 Functional Architecture

Figure 3-1 shows the functional architecture and is labeled the *Functional Coordinated Air and Space Operations System (FCASOS)*. There are five major components to *FCASOS*: a top level coordination controller (CC), three sub-controllers, and a battlefield simulation (SIM). For the purposes of this research, a *controller* (to include the term sub-controller) is defined as an entity whose job it is to monitor battlespace information and task or re-task resources when new information becomes available. The resources of CC are the three sub-controllers; the resources of the sub-controllers are the aircraft, satellites, and ORS assets. The frequency with which controllers are updated with new information is discussed in section 3.2.2.

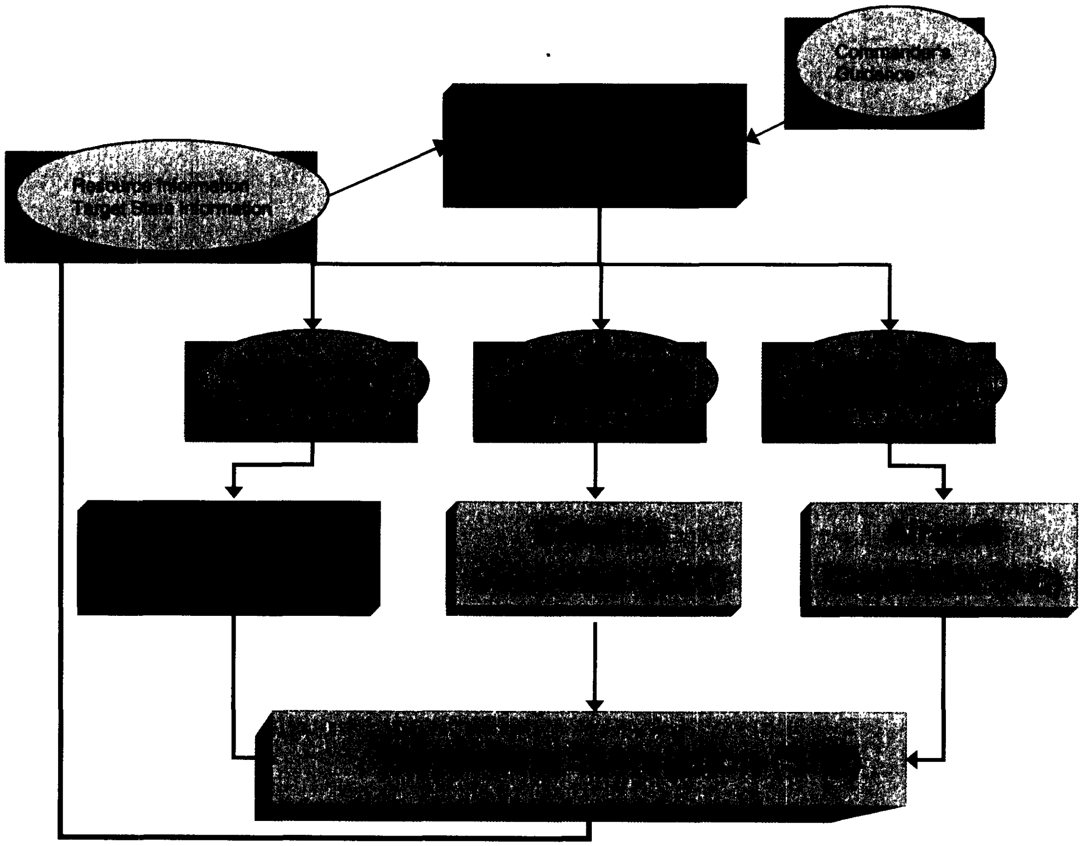

Figure 3-1: Functional Coordinated Air and Space Operations System (FCASOS)

The SIM in *FCASOS* is used to simulate, and evaluate the success of, controller-generated plans. The three sub-controllers include an aircraft controller (AC), a satellite controller (SATC), and an ORS controller (ORSC). The choice to use this type of structure is somewhat motivated by its hierarchical nature, which closely parallels the decision structure found in the military. Often, high-level commanders will make broad goals to direct the general path of the unit, while leaving the specifics to lower levels. In the CONOPS given in Chapter 2, the JFACC is in charge of all air and space resources. Here, the JFACC is modeled as the top-level commander, or CC. The hierarchical decision structure, then, involves CC producing high-level plans that serve as the inputs into sub-controllers. The three sub-controllers are in charge of resource tasking, and seek to assign their resources to targets in a manner that achieves the goals set forth by CC.

ORSC plans the launching and payload release of all operationally responsive spacelift resources (as outlined in section 1.2.2). SATC decides which satellites should look at which targets at what times. AC directs all aircraft in the campaign to look for and destroy the maximum number of targets on the battlefield. The SIM, then, takes the tasking from each sub-controller and simulates execution of these tasks in a given battlespace environment. In Figure 3-1, rectangles represent functional components and ellipses represent information flows. Red items represent models and software developed in this research (CC and ORSC), while blue items represent pre-existing software components. CC and ORSC are discussed at length in sections 3.3 and 3.4 respectively. An overview of the pre-existing AC, SATC, and SIM, is given in section 3.5. First though, it is important to understand system-level dynamics and how and when the functional components interact.

3.2.2 Planning and Re-Planning

In Chapter 2, the production of ATOs was highlighted, to include how new ATOs are generated on a 24 hour cycle. The reason that a single ATO cannot handle the planning for the whole campaign is because of the enormous amount of uncertainty inherent in battlefield operations. New targets emerge, old targets change location, strike missions fail to completely destroy their intended targets, and ISR missions return information that updates the IPB. To account for this uncertainty, strategic objectives, operational plans, and tactical missions must be updated on a frequent basis.

Define a *replan* as the output produced by one iteration of CC and one iteration of each of the sub-controllers. A replan involves one cycle through the functional system, starting with CC and ending with the SIM, and it completely updates the tasking of all resources in the battlespace. In air operations, there are three ways of determining how often to generate a replan. The first is to set a fixed interval of time such that new replans are generated after the specified time has passed from the previous one. The second way is to generate a replan only after a predefined event, such as a new high-value target appearing in the IPB. The third is a combination of the previous two, where replans are generated at the end of fixed intervals or after a predefined event, whichever comes first. We take the first approach – to re-plan at fixed intervals. It should be noted though that while the length of the replan interval is fixed for each SIM run, it may be changed from one run to the next. We define a SIM run as one complete battle campaign from start to finish.

In addition to deciding how often to replan, there is the issue of deciding for how long to plan. This is known as the *planning horizon*. The planning horizon is assumed to be for a length of time equal to or greater than the time of the next replan. Here, the concept of a rolling plan horizon is employed. At every replan, the piece of the previous plan that has yet to be implemented is cut off and a new plan is generated in its place. The main advantage of this approach is that it allows controllers to plan for the duration of the re-plan interval while taking into account the fact that the campaign keeps on going after the interval is over. This causes a smoother flow of resource assignment; there will be no surges in target assignments at the end of a replan interval in order to hastily gain leftover target value.

At the start of every replan, the SIM sends information to CC to provide an update of the state of the battle campaign. CC must determine how best to use resources over the planning horizon. It is this function of CC that is discussed next.

3.3 The Coordination Controller

It is the job of CC to manage the direction of the campaign through the collaborative tasking of sub-controllers. It is difficult to quantify what is meant by the term "collaborative." Instead of having one definition, we use the idea of differing *levels* of collaboration (or coordination), where a *level* is taken to mean a combination of 1) the amount of knowledge of sub-controller capability that is used by CC to make decisions, 2) the degree to which sub-controller decisions (or expected decisions) influence the plans of other sub-controllers, and 3) the degree to which joint objectives and constraints influence sub-controller plans. One extreme is no coordination, where CC merely passes sub-controllers whatever information is received from the SIM at each replan. The other extreme – complete coordination – is difficult to define and is not explored in this thesis. However, in a "highly" coordinated system, CC has knowledge of most sub-controller capability and can make decisions that effectively integrate those capabilities. We hypothesize that higher levels of coordination will produce increased target damages in shorter amounts of time and at a lower costs. The following sections further define what we term a "high" level of coordination by presenting the inputs, decisions, objectives, and constraints of CC.

3.3.1 CC Inputs

The inputs to CC come from the SIM and from user-specified parameters. Inputs can be broken up into three types:

1. **Commander's Guidance**
2. **Resource Information**
3. **Target Information.**

While CC receives updates to various parts of commander's guidance at different times, the resource and target state information is updated once at the beginning of every replan. Example inputs into CC are presented in Table 3-1. We devote a section to each of the three input types.

Commander's Guidance

Intent	Region 1	Region 2
WMD	1000	[illegible]
SAM	200	[illegible]
	Aircraft 1	Aircraft 2
Risk	0.01	0.4
s value	0.5	
	Target #	
TCT 1	45	
TCT 2	26	
	% destroyed	Time till end
End State	90%	4

Resource Information

Satellites	Orbital Elements	Sensor Type
Sat01	6631.0	EO
Sat02	6631.0	EO
ORS	Location	objective
SOV01	at base	--
SOV02	10.73, 120.32	tgt # 31, 32
Aircraft	AC01	AC02
Location	at base	[illegible]
Fuel	full	[illegible]
Armament	full	full
Objective1	--	hit tgt 20
Objective2	--	ISR tgt 21

Target Information

Attributes	Target #1	Target #2
Location	[illegible]	[illegible]
Certainty	[illegible]	[illegible]
Time Observe	7200	3600
Target Type	WMD	SAM
Certainty	90.00%	70.00%
Damage	0.5	0.75
Certainty	40.00%	30.00%
Region	1	3

Table 3-1: Sample Inputs into CC

Commander's Guidance: In commander's guidance, the user is able to quantify military objectives for the campaign. Five components to commander's guidance are examined below. Commander's intent, which is the first component, is provided at the start of every campaign phase and is given in the form of a numerical measure for every region of the battlefield. It measures the relative importance of destroying targets of different types and within different regions. Developing a good method of scaling and quantifying intent is a difficult problem not addressed here. We assume an acceptable scale has been found and appropriate values have been given exogenously. These intent values are seen as independent from one another in the sense that they only measure a target's worth in-an-of-itself, not how that target might contribute to other military objectives. For example, on a scale of 1-1000, a WMD factory (WMD) in a particular region might have an intent value of 1000 while a SAM site (SAM) in the same region might have an intent value of 200. In essence, this means with the option of destroying five SAMs or one WMD, CC is indifferent about the decision. If there is an entity whose destruction would hinder campaign objectives, then its intent should be a negative value.

The second component in commander's guidance is the acceptable risk level to aircraft, given by aircraft type, and in the range [0,1]. This is a measure of the maximum single-aircraft attrition probability that the commander is willing to accept for any given mission. For instance, if a global hawk is given a risk level of 0.80 and an F-22 is given a risk level of 0.01, then global hawk could potentially perform missions in which they are 80 times more likely to get shot down than an F-22.

The third component in commander's guidance, labeled s, is a measure of the importance of speed vs. cost in the campaign. Cost includes expected aircraft attritions and expected operating cost. The parameter s is a weighting in the range [0.0, 1.0]. If s = 0.0 then there is no value given for a speedy campaign – the objective is to destroy the maximum amount of targets at a minimum cost, no matter how long it takes. in contrast, if s = 1.0, no value is given for resource conservation – the objective is to destroy the maximum amount of targets in the minimum amount of time, regardless of cost. It is important to note that the relation of s to cost and speed is not linear. As s increases from 0, the campaign length decreases exponentially while the cost increases exponentially. This relationship is shown in Figure 3-2.

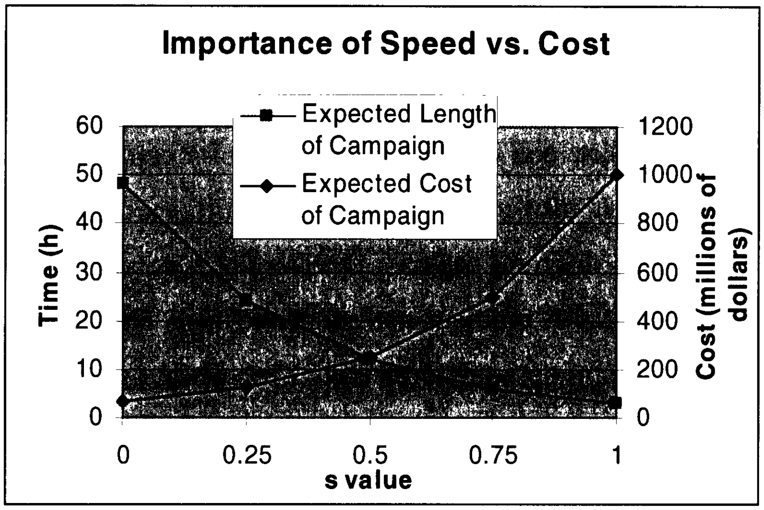

Figure 3-2: Importance of Speed vs. Cost Parameter

The last two components of commander's guidance are a list of all time-critical targets and an updated campaign end state. If a target is flagged as time-critical, CC tasks the sub-controllers so that the nearest available resource (in terms of time-on-target) be assigned that target, regardless of all other factors. The campaign end state, for our purposes, is in one of two forms. It is either a desired percentage of targets destroyed or a specific amount of time until the end of the campaign. In either case, it is a quantifiable measure of the end state desired by the commander.

Resource Information: In order to be able to coordinate among sub-controllers, CC needs information about the state of the resources. Some of the resource information need only be provided once, at the start of the campaign. For ORSC, this is the number of SOVs that are in inventory, the possible SOV configurations of CAVs/microsats, the cross-range of CAVs, and the time required to launch an SOV, complete one of its two possible missions, and return ready for another launch. For each satellite in the satellite constellation, initial resource information includes all orbital elements except mean anomaly, which is updated at every replan. Also given for each satellite is the type of sensor on board, field of view, maximum slew, and the slew rate. For aircraft, CC knows the weapon and ISR capabilities of each aircraft, initial locations, cruising speed, fuel capacity, etc.

While static resource information is provided only once, the current state of dynamic information is available to CC at each replan. For ORS assets, dynamic information involves the location and target objective (if on a mission) of all SOVs as well as the number of SOVs at base ready for launch. For satellites, the mean anomalies are the only set of orbital elements that change with time and so are considered dynamic information. For aircraft, CC is passed the location, fuel and armament information of each aircraft as well as the target objective of each aircraft (if on a mission).

Target Information: The *imperfect target state information* assumption is modeled by the fact that CC has only probability distributions of target attributes. CC keeps these attribute distributions in an information model that is updated at every replan. The information model is similar to IPB and includes point estimates along with certainty estimates. For instance, the best estimate of a target's location might be [-5.00, 142.25], given in lat/lon coordinates and with 50% certainty. After an ISR mission is sent to find that target, this estimate could be updated to [-5.01, 142.27] with 90% certainty. The certainty should then start to degrade with time until another ISR update is received. The target attributes given to CC as inputs are target location (latitude and longitude), time of the last observation, target type, observed damage state, and the appropriate region of the battlefield. An even more precise way of capturing uncertainty in the information model, which is not adopted here, would be to use Probability Mass Functions (PMFs) for every attribute.

3.3.2 CC Decisions

CC sends directions to the sub-controllers at the beginning of the campaign and after every replan. These directions help to guide sub-controllers in the tasking of their resources. Whereas sub-controllers decide *how* to process their assigned targets with their given resources, it is the job of CC to determine *which* targets to strike and ISR, and with what timing. CC therefore does not assign *specific* resources to targets, but rather assigns the *sub-controllers* to targets. CC passes down the following information:

1. **Intelligence Preparation of the Battlefield (IPB) – target state information, geographical areas of interest, etc.**
2. **Target Commands (ISR target #1, strike target #2, do nothing to target #3, etc.)**
3. **Timing of Prescribed Target Commands**
4. **Relative Valuation of each Target (including if target is Time-Critical)**

While the IPB is determined by the information model in CC, the last three items on this list constitute decisions. These decisions are discussed below and examples are provided to demonstrate their impact on the campaign. Figure 3-3 is an illustration of the information flow from CC to the sub-controllers.

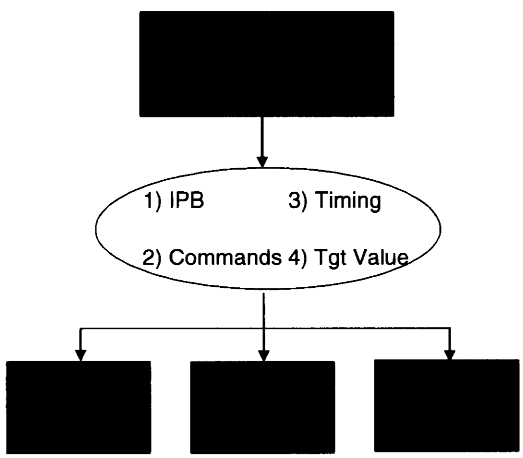

Figure 3-3: CC Information Flow

Target Commands: Although each sub-controller should have the capability to function independently of CC, effective coordination among sub-controllers is aided when target commands are given by CC at every replan. If CC were to hand down merely a target list with no commands, or instructions for each target, sub-controllers would attempt to process targets completely by themselves, thus preventing sharing of sub-controller functions for targets. Conversely, by requesting that specific actions be taken against certain targets, CC could, for example, have AC attack a target and SATC perform the BDA on that target shortly thereafter.

Timing of Prescribed Target Commands: Timing is vital to our ability to have effective coordination. To illustrate this, consider a situation where an AC strike asset has the ability to attack a particular target fairly easily (as would be the case if the aircraft is near the target and the target is not a threat, not well-defended, etc.). Secondly, suppose that the nearest AC ISR resource is hundreds of miles away. Also, a satellite will be directly over the target by the time an AC strike mission could arrive. This scenario is shown in Figure 3-4.

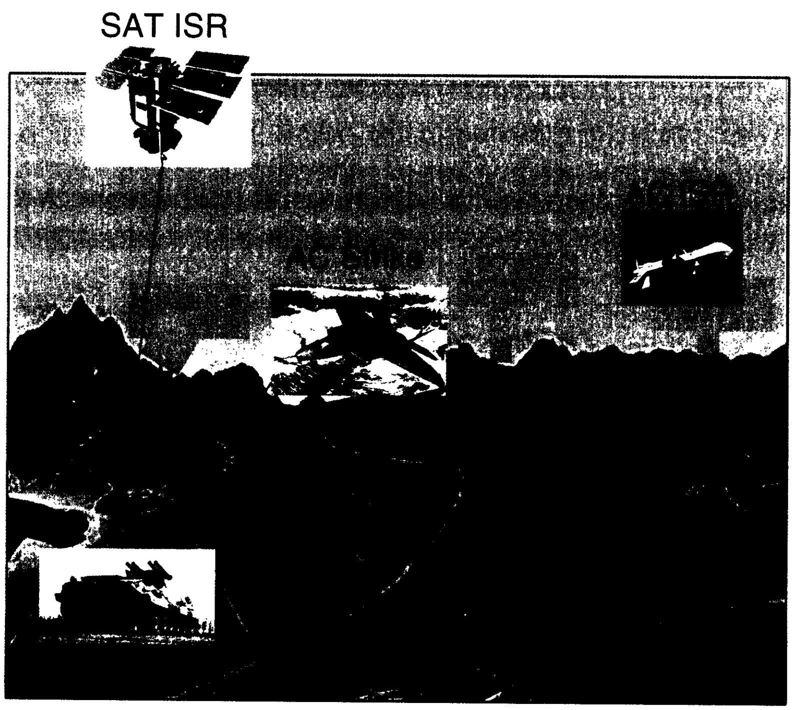

Figure 3-4: Strike/BDA scenario

Tasking AC to strike the target and SATC to perform BDA (each at a specific time) should cause a strike/BDA combination that happens faster and in closer succession than would otherwise be possible. Moreover, this would allow the unused ISR aircraft to be free to do something closer to its current location.

The *level* of coordination, as defined in section 3.3, dictates how close together such actions may be performed. For instance, suppose the sub-controllers have a constraint that they must perform the prescribed action inside a 30-minute window (where the center of that window is the time given by CC). Then clearly, CC would not want to task two sub-controllers with processing the same target closer than 30 minutes apart. Otherwise, situations could arise where the BDA comes before the target is struck.

Relative Valuation: The relative target valuation has a similar meaning to the commander's intent. It is a numerical value given to each target that measures its importance to CC relative to all other targets. However, it differs from commander's intent in the following way: It is not only a measure of target worth, but also incorporates the value in leading to the accomplishment of CC's campaign objectives, which are discussed in the next section. Consider an undestroyed SAM and an undestroyed command and control facility (C3). If the location of the SAM is very close to that of the C3, the SAM could potentially shoot down aircraft on the way to destroy the C3. One way for CC to mitigate this is to decrease the AC relative valuation of both targets while increasing the ORSC relative valuation of the SAM. Thereby, ORSC is encouraged to strike the SAM while AC is encouraged to remain clear of those targets, at least until the next iteration. This potentially prevents an aircraft attrition, still causes the SAM to be destroyed, and frees up aircraft for other less-protected targets. Moreover, doing this allows a way around the *Target State Independence* assumption. Instead of linking target values together in the objective function of CC, it links them together in the inputs to the sub-controllers.

Lastly, to gain the most benefit out of coordination, there needs to be some element of feedback from the sub-controllers to CC. At least two reasons for this are apparent. First, suppose CC passes a strike objective to ORSC, where it is then determined to be infeasible. ORSC might decide that with slight alteration to the requested time of strike, the target objective could be feasible. If ORSC could relay this information, CC could alter the time of strike, as well any other sub-controller's plans that were contingent upon this strike. In this way, a greater level of coordination is achieved. The second reason is similar. Suppose one of the sub-controllers could do more than is requested of them at a particular iteration, without significant increase to cost. If the sub-controller could inform CC, then another, more efficient plan using all of that sub-controller's resources might be found. Once again, the decisions produced by CC are for the purpose of coordinating in a way that accomplishes the three main campaign objectives.

3.3.3 CC Objectives

In keeping with the CONOPS of Chapter 2, the objectives of CC include:

1. **Maximize Weighted Expected Target Damage**
2. **Minimize Expected Cost**
3. **Minimize Expected Time to Completion**

The primary objective is to Maximize weighted expected target damage. Minimizing expected cost and expected time to completion are secondary and, as discussed previously, the commander's guidance governs the relative importance between these two. Also, the objective function is confined to a linear form because of the assumptions: *Value and Constraint Proportionality* and *Value Additivity*.

Maximize Weighted Expected Target Damage: This is the first and most important functional objective. It satisfies two operational goals listed in Chapter 2 – first, to destroy all targets deemed "high value" by the commander and second, to establish air superiority over the battlefield. At the beginning of the campaign, CC receives the Commander's Intent, as described in section 3.3.1. The intent is used as a measure of relative value in comparison to all other targets. The idea is this: if the intent lists a WMD as being five times more valuable than a SAM, then an expected WMD destruction should be worth five times as much as an expected SAM destruction in the objective function (disregarding cost and time for now).

The target damage objective causes two problems. First, in a real battlefield situation, targets are often only partially destroyed. The *Finite and Discrete Attributes* assumption precludes the possibility of representing an infinite number of possible target damage states. The damage state space must be discretized. A two-state representation of target damage could be *{live, destroyed}*. Similarly, a three-state representation might be *{live, 50% damaged, destroyed}*, and so on. The second problem is that some targets are reconstitutable, meaning enemy troops can perform maintenance and cause the target to transition from a destroyed to a damaged state, or from a damaged to a live state. The issue is that if a low-value target has a very short reconstitution time, CC might waste resources destroying it over and over in order to gain the target's value again and again. We assume here that only high-value targets are reconstituted.

Finally, we point out that the tasking of ISR missions are listed as constraints instead of objectives. To gain the value for the expected destruction of any target, certain ISR requirements must be satisfied. These constraints, discussed in section 3.3.5, cause the objective function to *implicitly* include ISR goals because, in order to gain any value for target damage, the target must be ID'd before strike and BDA'd after strike.

Minimize Expected Cost: It is difficult to measure the expected cost of air and space operations. There is more than one component of cost and it can be problematic to try to place components on the same scale. How does the cost of losing a pilot's life from an aircraft attrition compare with the operating cost of a mission? This is a hard question to answer. Nevertheless, both of these costs must be taken into account. The two categories of costs considered here are aircraft attrition cost and operating cost. Aircraft attrition cost includes both the value placed the loss of human life and the monetary cost of the aircraft that was lost. Operating cost includes the cost of fuel, maintenance, and any released weapons. Collateral damage on civilian infrastructure or population is not included here. Instead, entities that the commander wishes not to destroy are simply given a negative value in the Commander's Intent.

Minimize Expected Time to Completion: The expected time to completion can either be a fixed amount of time or the expected time until a specified total target damage percent is reached. In the latter case it quantifies the speed with which the target damage objective is accomplished. If the commander places speed as a higher priority than cost then two things should happen: First, aircraft should perform more risky missions – attempting to strike a well-protected high-value target before clearing a pathway to it is one example. Secondly, ORSC should use higher numbers of CAVs on each of its assigned targets (versus conserving resources) to ensure complete destruction. This type of strategy has, we hypothesize, a higher rate of total target damage but at the cost of heavy losses and resource consumption. Placing time to completion as a lower priority than cost would then produce the opposite effect.

3.3.4 CC Constraints

CC should consider sub-controller constraints when generating commands to send to the sub-controllers. It is true that CC does not have full knowledge of the capabilities and limitations of the sub-controllers; CC cannot, for instance, analyze all the different routing options for a particular aircraft mission. However, with an accurate *estimate* of sub-controller capability, constraints can be modeled so that sub-controllers do not get tasked with a mission or a combination of missions that they are not capable of handling. Constraints include:

1. **Resource Capacity Restrictions**
2. **Resource Performance Constraints**
3. **ISR Requirements**
4. **TCT Forcing Constraints**
5. **Integrality Constraints**

Resource Capacity Restrictions: It is assumed there are only a limited number of aircraft, SOVs, and satellites compared to the relatively large number that would be required to destroy 100% of the targets in a single planning interval. The purpose of this constraint is to attempt to tailor the target allocation to the capabilities of sub-controllers. If this constraint were not included, the workload among sub-controllers might get unbalanced, where one sub-controller is inefficiently trying to handle a majority of the targets. In order to formulate this constraint, there needs to be some measure of how many of each type of resource it takes to do each type of mission.

Resource Performance Constraints: The resources belonging to sub-controllers are constrained by the laws of physics and their own capabilities. These constraints are related to the resource capacity constraints – it depends on the level of detail sought in the CC formulation. The purpose of resource performance constraints in the CC function is, once again, to make sure sub-controllers do not get more than they can handle. An example is given for illustration. If, at the current replan interval, an LV has just released a CAV and is beginning its descent into the ocean where it will then be recovered, a resource performance constraint is needed to ensure the LV has a chance to get back to base before reassignment. The constraint here is an expected amount of time based on LV recovery characteristics.

ISR Requirements: The approach toward the ISR function is that it is completely in support of strike operations. This means that to gain any target value on a strike mission, certain ISR requirements for that strike mission must be met. The target must be properly identified, and once struck, BDA'd. This is enforced by listing two ISR requirements – identifying and damage-assessing – as constraints on CC. Thus, any strike mission tasking should generate a series of other missions that perform the two ISR functions.

TCT Forcing Constraints: Part of the commander's guidance is a listing of Time-Critical Targets (TCTs). These are targets of the highest priority and everything should be done to cause their immediate and total destruction. For every target listed as a TCT, a constraint is generated on CC that requires a strike mission to be immediately undertaken, even if pre-strike ISR requirements are not met.

Integrality Constraints: There is a chance in a non-integer formulation that CC might assign fractions of one mission function to different sub-controllers. The different functions are **1) pre-strike ISR, 2) strike, and 3) post-strike BDA**. For example, CC might designate both ORSC and AC to inflict 50% damage upon the same target. Although in theory this might be feasible, limitations to the real-world scenario prevent it. There is not enough coordination between sub-controllers to effect this option. The resulting constraint, then, is that the fraction of each function assigned to each sub-controller must be either 0 or 1. Under this constraint, different sub-controllers may perform *different* functions on the same target. SATC may perform ISR on a target while AC may strike the target. Also, sub-controllers are allowed to attack a target that was attacked by a different sub-controller in a previous planning interval.

3.4 The ORS Controller

The ORS controller, or ORSC, is in charge of tasking re-usable launch vehicles. Although a specific type of launch vehicle – the SOV – is discussed in Chapter 2, the generic name "launch vehicle" (LV) will be used for the remainder of this chapter. The LV can carry two types of payloads: entry vehicles (EV), capable of strike, or microsatellites, capable of ISR (or a combination of both). Once an LV is launched, it delivers its payload, circles the earth, and lands in the ocean. It can be recovered and made ready for re-launch within a matter of hours.

3.4.1 ORSC Assumptions

In the context of the functional system, the following assumptions are made regarding the operation of ORSC.

First, the ORSC re-plan interval and planning horizon (as well as those of the other two sub-controllers) is confined to be the same as that of CC. In this way, sub-controllers will only re-plan after a CC replan.

Secondly, there are only a limited number of EVs relative to the number of targets on the battlefield. So while ORSC might aid in striking and performing ISR on certain targets, ORSC is not, by itself, capable of destroying all of the targets on the battlefield. Also, all LVs in the ORSC system come from a single base.

Thirdly, each EV is limited to carry exactly one munition that can strike a single target. The extent of damage on this target depends on the type of target and the proximity of the EV to the target at impact.

Fourthly, a microsat, once launched in an appropriate orbit, can provide timely ISR of a target. Once in space, the microsat joins the satellite constellation to supplement the capacity and capability of SATC.

Finally, all modeling assumptions made in section 3.1 Modeling Assumptions apply to ORSC. To summarize, **the abilities of ORSC include 1) striking a target, 2) performing a one-time ISR of a target, and 3) augmenting the ISR potential of SATC.**

3.4.2 ORSC Inputs

The inputs to ORSC come, with one exception, directly from CC. Inputs from CC include: 1) a target list with defined objectives for each target, 2) the timing requested for each target objective, and 3) the weight, or relative value, given to each target. Lastly, ORSC has access to IPB, which has current estimations for the attributes of all targets on the battlefield.

Even though CC has its ISR objectives modeled as constraints, this approach is not taken with ORSC. Consider what might happen in the following situation: Suppose at a particular time in the campaign, there are numerous targets requiring ISR and a few, high-value targets, requiring strike. The high-value targets are heavily protected so that only ORSC can effectively strike them. If ORSC is constrained with ISR requirements, the ability to aid in strike is negated. Rather, ORSC should be allowed to decide what combination of strike and ISR missions produce the highest value. It should be noted that the target weighting (given by CC) for strike objectives must be on the same relative scale as that of ISR objectives. Target weight reflects the relative importance to the commander for doing the associated action.

One addition to these inputs is related to the existing satellite constellation. ORSC needs some measure of how valuable it would be to have the satellite constellation augmented with an extra microsat. This microsat "potential" value is a function of two things. The first is the remaining number and capability of satellites and ISR-aircraft. If existing resources do not have the capability to meet future ISR demand, an extra microsat would be valuable. Total ISR capability can be expressed as the average number of targets on which a resource or subset of resources can perform ISR per unit time. For instance, satellites might be able to perform ISR, on average, on three targets per replan interval. The second part to microsat value is the expected number of ISR missions remaining in the campaign.

3.4.3 ORSC Decisions

ORSC's goal is to assign resources to both ISR and strike missions in such a way as to follow the objectives of CC, as well as certain tactical objectives. At every re-plan, ORSC must generate resource tasking information by making the following decisions:

1. **Number and configuration of tasked LVs**
2. **Launch time and inclination of each tasked LV**
3. **Target assignment for each EV**
4. **Release time and target assignment for each microsat**

No differentiation is made among LVs until one is configured for launch. It is assumed that any given LV may be configured in any possible launch configuration.

Number and configuration of tasked LVs: ORSC must determine how many LVs to launch, and in what configuration. Because of the assumption that EVs and microsats are relatively cheap compared to a launch, each LV will either launch with a full payload capacity or not launch at all. If a launch is going to occur, it only makes sense to fill up capacity to gain all potential benefit. For a case where there are 10 LVs, ORSC may choose to launch between zero and 10 completely full LVs.

Define a launch option as a list of LVs to launch along with the configuration type of each LV. The total number of launch options is calculated for the 10-LV after listing some assumptions: First, each LV can hold any combination of four microsats or EVs. Every LV may then be configured in one of five possible ways (e.g. 1 EV and 3 microsats). Secondly, because LVs with the same configuration are indistinguishable, the configuration ordering does not matter.

The total number of launch options is calculated by way of an illustration: Consider each launch vehicle as a separate "bin." Allow each bin to hold some number of tokens from 1 to 6, where the number of tokens represents the configuration type.

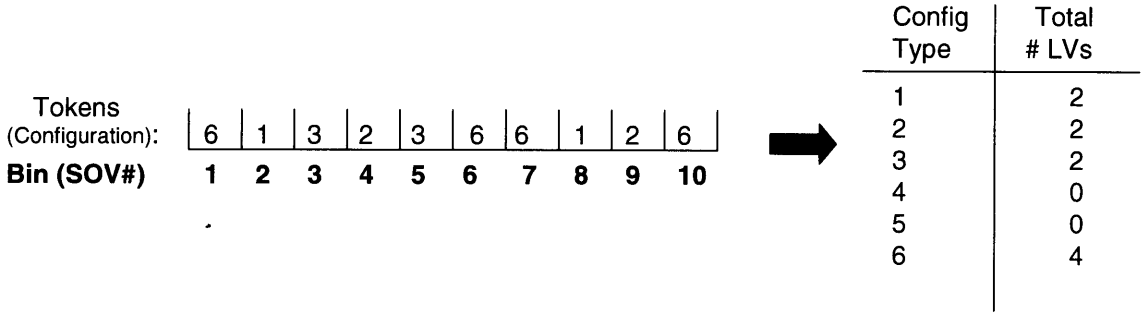

Figure 3-5: Illustration of Launch Configurations

Configuration type 6 is added to represent the empty configuration, or no launch of that vehicle. So, if bin 2 contains 4 tokens, this means the LV #2 has configuration type 4. If bin 3 contains 6 tokens, LV#3 is not launching. Figure 3-5 shows a case where two vehicles have configuration type 1, two have type 2, two have type 3, and four are not launching at all.

Counting the number of possible launch options first requires a slight alteration to Figure 3-5. Five "dummy" bins are added (to make 15 total bins). Five "dividers" are then placed into five of the bins. Dividers serve to partition subsets of LVs with different configurations. Thus, all bins to the left of the first divider will have configuration type 1. All bins between dividers one and two will have configuration type 2, etc. Figure 3-6 is the new representation of the case presented above.

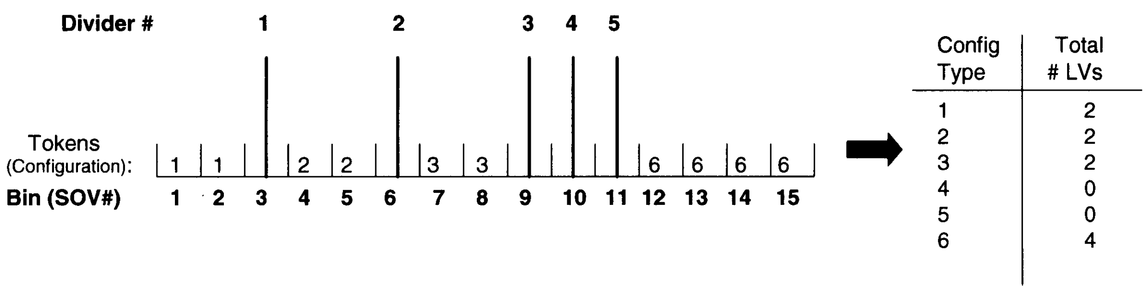

Figure 3-6: New Representation of Launch Configurations

Using this new representation causes each divider to occupy a different bin. Also, it is important to note that the configuration of launch vehicles is completely determined by the *location* of the dividers. For instance, in the figure above, if the first divider were to move one bin to the right, there would be one more type-1 vehicle and one less type-2 vehicle. Now to determine the number of launch options, we simply count the number of ways it is possible to place the five dividers in the 15 bins. As it turns out, this is the number of combinations of 5 out of 15, or "15 choose 5." This yields a total of 3003 launch options (including one option of no launches). For the generic case, let L = the number of available LVs and C = the number of configuration types (including the empty configuration). Then the number of launch options, N, is given by:

$$N = \binom{L + C - 1}{C} \tag{3.2}$$

Launch options are then decided using their expected profitability, which is based on: a) the capability of the resources being launched and b) the target weighting, and c) the result of other ORSC decisions.

Finally, LVs are reusable once they are recovered from the previous launch and at some replans, one or more LVs will return to base and become ready for launch within the course of the subsequent planning horizon. ORSC must take this into account when determining LV assignments.

Launch time and launch inclination of each tasked LV: The LV launch time and inclination determines both CAV strike windows and microsat ISR windows. The launch time is defined as the physical time of liftoff. The launch inclination is defined as the angle that the projected orbital plane makes with the equator. Once liftoff has occurred, the LV accelerates to the predetermined speed necessary to either do a once-around of the earth or to attain LEO (if there are microsats on board, the LV needs to attain LEO to release them). At the payload release time, which is also predetermined, the LV releases any CAVs and microsats, de-orbits, and falls into the ocean to be recovered. The CAVs and microsats, then, have strike and ISR windows depending on when they were released. Each window has as its bounds the minimum and maximum time that the resource is physically capable of performing the requested action.

Target assignment for each EV: Two factors go into making the EV target assignment: the target damage potential and the result of other ORSC decisions. First, target damage potential is discussed. If it only took one EV to completely destroy any target, ORSC could simply assign EVs according to a value-ordered list of targets in which each EV is assigned to the target remaining with the highest value. As it stands, EVs have a certain probability of destruction based on target type, region, and number of EVs. For instance, it might be given that three EVs used together have a 0.75 probability of destroying a WMD target in region #1. Logarithmic interpolation can be used to determine the probability of destruction for numbers of EVs other than what is given (three in this case). Thus, two EVs would have a 0.559 probability of destroying the WMD.

In addition, the target damage state must exceed a certain damage threshold (determined by the commander) to be considered destroyed. So, if the damage threshold of the WMD above is 0.5, then ORSC might be content with only assigning two EVs. This can have the following effect: if a high-value target has a very low probability of destruction but a high threshold requirement, ORSC might forsake it for an easier-to-destroy but lower-value target. However, in the case of targets listed as time-critical, the ORSC objective function is given a high enough value for the TCT so that ORSC choses to strike the TCT over any other action, if feasible.

The second factor in determining the ORSC target assignment involves the other ORSC decisions. Consider the launch configuration. If an LV is configured to hold only two EVs, ORSC should not assign it to a target requiring three EV hits. Or if it does, it should launch another LV in close succession to ensure the target's destruction. Time of launch and launch inclination are also important factors to the target assignment. These two quantities help determine target strike time and it is only desirable to strike a target within its requested strike window. Therefore, highest precedence should be given to targets whose requested strike time falls within the EV strike window.

Finally, it is assumed that all payloads on the LV must be released at the same time. The release time should be such that the average time difference between requested and projected strike time be minimized.

Target assignment and release time for each microsat: Just as each EV is assigned a target to strike, each microsat is assigned a target for ISR before joining the satellite constellation. When tasked with an ISR objective, there is a certain probability that the microsat will identify or assess target damage correctly based on target type, region, sensor type, and possibly weather. All microsats are assumed to have the same sensor type. A target that is high-value but has a low probability of correct ISR could be forsaken for a lower-value target in some instances. Also, targets should only be assigned to microsats if the time that the microsat will be over the target falls within the target requested look window. In the case of a TCT, the value of the target should be so great that the TCT is assigned over any other. Once joined in the satellite constellation, control of the microsat transitions to SATC.

3.4.4 ORSC Objectives

The objectives for ORSC constitute a) strategic goals to fit in line with what CC sends down, and b) tactical goals and considerations that would only be of concern to a lower-level commander. The ORSC objectives are:

1. **Maximize Weighted Expected Target Damage**
2. **Maximize the Time Proximity to the Requested Target Objective Time**
3. **Maximize Mission Readiness**
4. **Minimize Expected Cost**

All four objectives here have strategic as well as tactical elements. Also, while the target damage objective is likely the most important, the specific scenario, including commander's guidance, dictates the actual difference in importance among objectives. Once again, a linear objective function is used.

Maximize Weighted Expected Target Damage: Each target has a weight given by CC and a probability of destruction given by the initial scenario data. The probability of destruction is based on target type, target region, and number of EVs used. It can also include the launch inclination of the LV and the release time of EVs that are aiming for the target. Thus, for any launch option, EV-target assignment, and series of launch inclinations and release times, the weighted expected target damage is calculated as follows: The sum is taken over all assigned targets and weighted by the probability of destruction based on all components listed above as well as the target weight given by CC. The objective, then, is to find the launch option, EV-target assignment, and series of launch inclinations and release times that provide greatest weighted sum.

Maximize the Time Proximity to the Requested Target Objective Time: At each replan, CC provides ORSC with requested times to take action against each target. This includes either a requested strike time or a requested ISR time, depending on the objective. In order for effective coordination among sub-controllers to take place, ORSC must do everything possible to ensure the actual strike or ISR time is as close to what was requested as possible. Suppose a SATC is tasked with performing BDA on a target exactly 15 minutes after the requested ORSC strike time. If ORSC is as little as 15 minutes late with its strike mission, the result could be a missed BDA. A parameter with a bell-shape curve is used on the time-proximity of requested strike time to expected strike time. There should be a very narrow window in which ORSC could gain nearly all the strike value.

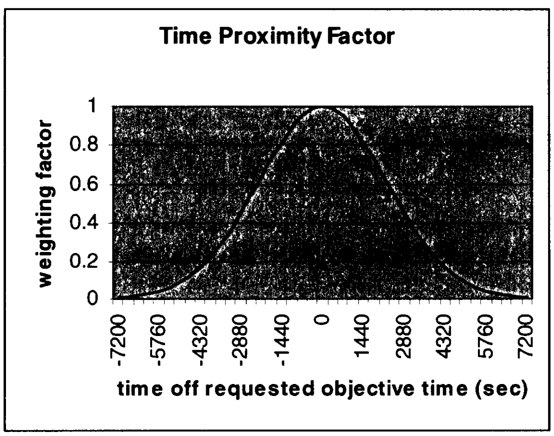

Figure 3-7: Example Time Proximity Factor Weighting

However, as expected strike time gets further and further away from requested strike time, the parameter should fall off sharply, and then level out towards infinite. The expected strike time is a result of the launch time, launch inclination, and release time. The resulting curve for the time-proximity parameter is shown in Figure 3-7.

Maximize Mission Readiness: Mission readiness is defined here as the ability of ORSC to handle as many of CC's tasks as possible. This translates into the percentage of LVs available for launch. The only decision in the model that causes change to the ORSC mission readiness state is launching LVs. Thus, if ORSC kept all LVs on the ground for the duration of the campaign, the mission readiness state would be at 100% the whole time. Of course, no targets would get struck in that case. It follows that a high mission readiness state and high target damage are strictly competing objectives. If ORSC has a very high probability of getting no new tasks between now and the time it takes an LV to be launched and recovered, the mission readiness objective is redundant and should be given a low importance. If, however, ORSC expects several high-value targets to be tasked within the next few re-plans, ORSC must weigh the benefit of launching LVs now versus having them around to handle future targets.

Minimize Expected Cost: The cost of microsats and EVs – namely the cost to manufacture and deploy them – is assumed to be dominated by the cost of an LV launch. We therefore assume that there are sufficient EVs and microsats available for any ORSC plan. The objective of minimizing cost is then translated into minimizing the number of LV launches. This is another objective that directly competes with maximizing target damage.

3.4.5 ORSC Constraints

The constraints on the ORS Controller include:

1. **Resource Capacity Restrictions**
2. **Resource Performance Constraints**
3. **Integrality Constraints**

Unlike CC, ORSC does not have any constraints that require ISR or strike to be performed on a particular target. While CC could assign as many targets as it wants to the sub-controllers, the sub-controllers themselves are limited by their resources. ORSC cannot task a mission if it has no available LVs. Therefore, both ISR and TCT objectives (given by CC) are put in the ORSC objective function. In this way, ORSC is constrained only by physical limitations of resources.

Resource Capacity Restrictions: There are two sets of resource capacity constraints. First, ORSC cannot launch more LVs than what is available at any time period. Available LVs include those that are at base currently and those that will return from a mission before the next replan interval. Because the ORSC planning horizon will likely be shorter than the launch and recovery time of an LV, ORSC does not consider the possibility of launching an LV, recovering it, and launching it again. It does take into account those LVs that were launched on a previous re-plan and are expected to return within this planning horizon. The number of LVs available at any time period is given by:

$$numLVavailable = numFromLastPeriod - numLaunched + num\mathrm{Re}\,\mathrm{cov}\,ered \qquad (3.3)$$

The second capacity restriction deals with what each LV can carry. Any given configuration has limitations on the number of EVs and microsats an LV can hold. For instance, configuration #3 might limit an LV to carry no more than two EVs and two microsats.

Resource Performance Constraints: It is assumed the LV launch time and inclination completely determine the strike and ISR windows. A "window" is defined as a contiguous set of time periods where the requested action is feasible. The LV payload is then constrained as follows: 1) the EV-target assignment must be scheduled to take place during the strike window, and 2) the microsat-target assignment must be scheduled to take place during the ISR window. For example, suppose a launch is planned for time 1800, with a resulting window of feasible strike times equal to [7200, 9000]. This constraint set requires all EVs to have target assignments scheduled to take place in the window [7200, 9000]. Also, LVs carrying microsats must be launched with an inclination that enables the microsat to reach orbit and join the satellite constellation. Lastly, the release times of all payloads on an LV are constrained to be at the same time.

Integrality Constraints: The final set of constraints for ORSC is simply that only an integer number of resources may be assigned. The number of assigned LVs, EVs, and microsats cannot be fractional.

3.5 Existing Functions

There are three functional pieces that have not yet been discussed – the Aircraft sub-controller (AC), the Satellite sub-controller (SATC), and the Battlespace Simulation (SIM). For the implementation of the functional architecture that is given in Chapter 4, pre-existing software that models these three components is available. The existing software is quite complex and would require a substantial amount of time to duplicate. Incorporating a large amount of existing code into the system allows for a greatly increased modeling capability. This benefit outweighs, in our estimation, the cost of integration as well as the cost of not being able to build AC and SATC exactly to fit into the functional coordination architecture. We discuss the purpose of AC, SATC, and the SIM below, and go into more depth about the actual systems in Chapter 4.

AC: It is the responsibility of AC to assign both ISR and strike aircraft with: 1) target objectives; 2) routing to get to each target; 3) weapons and sensors; and 4) specific sets of aircraft to use for each mission. The overarching goal of AC is similar to that of CC, except using only aircraft resources. One interesting feature of AC is that it allows CC to specify the tolerable risk for any mission and for any aircraft type. If AC generates a possible mission where the risk of at least one tasked aircraft is greater than the tolerable risk, the mission will not be executed.

SATC: The satellite sub-controller reads a weighted list of targets with each target's value and lat/lon coordinates. SATC determines which of its satellites are capable of seeing any targets over the course of the planning horizon. Under normal circumstances, with a small battlefield area (e.g., 40,000km^2) and with an average size constellation (e.g., in the 10s of satellites), only a small percentage of satellites are able to see any targets at all in the course of a planning horizon. SATC tasks satellites in such a way that optimizes the total value gained based on the individual target values. SATC helps the objectives of CC by providing risk-free ID and BDA of targets if BDA occurs sooner using the satellites, and more ISR aircraft are freed up. Also, AC and ORSC strike resources will be used more efficiently, thereby better accomplishing CC's objectives.

SIM: The purpose of having a simulation is to provide a closed-loop evaluation of controller-generated plans in a simulated battlespace environment. SIM is a discrete event simulation developed by Draper Laboratory that we use for this purpose. It reads in planned tasks from sub-controllers at fixed intervals and simulates the execution of these tasks in the battlespace. It models all satellites, ORS assets, aircraft, targets, sensors and weapons and the interactions among them.

Three main areas of probabilistic uncertainty are modeled in the SIM. First, there is the uncertainty involving initial target location. At the beginning of the campaign, the lat/lon coordinates of targets are sampled from uniform distributions. Different distributions are used for targets of different type. Depending on the scenario, controllers might have either complete or only partial knowledge of target locations.

Secondly, there is uncertainty involving state estimation, which is contained in the IPB. The indirect purpose of every ISR mission is to update the target and threat knowledge in the IPB. However, the IPB is never completely accurate. Even specific data brought back from ISR missions include inaccuracies.

Thirdly, there is the uncertainty involving damage to resources. When passing through a high-threat area, each aircraft has a non-zero probability of being shot down. Different factors influence this probability, such as whether or not there is a "jammer" aircraft in the area to disrupt the threat's radar signal.

Lastly, we point out that the target damage in this simulation is not uncertain. For a given weapon task against a target, the resulting damage is fixed. Now that all functional components in *FCASOS* have been addressed, we proceed to introduce algorithms for CC and ORSC in the next chapter.

4 Algorithms and Formulations

There are five components of the functional architecture that are formulated mathematically and coded into software. These include the top-level CC, the three sub-controllers (ORSC, AC, and SATC), and the simulation, or SIM. First, a literature review of various formulations for the planning of air operations is provided. Then, the software set is discussed and an overview of each pre-existing piece of software is given. Finally, algorithms for the two new components – CC and ORSC – are presented.

As a motivation for this chapter, the three primary goals of this research are reiterated:

1. **Formulate a coordination algorithm for CC that causes sub-controllers to work collaboratively**
2. **Formulate an assignment algorithm for ORSC to task resources in a way that achieves CC objectives**
3. **Quantify the benefit of integrating air operations with satellite and ORS operations**

The first two are addressed in this chapter; the third is the subject of Chapter 5. There are three algorithms for CC in this chapter, each with a successively higher level of coordination among sub-controllers. There are also two ORSC algorithms, including one heuristic and one optimizing. Everything except the optimizing version of CC is implemented and tested on the SIM. Before discussing the algorithms, it is important to gain an understanding of what has been done in previous air operations planning models.

4.1 Literature Review

In order to solve problems similar to *ASOPP* (presented in Chapter 2), a number of different types of formulations have been proposed. These formulations can be grouped into three categories – static, dynamic and a combination of both. A static model is labeled as such because the solution does not change with time. Although the model might consider expected future occurrences, the final solution is not contingent upon them. On the other hand, a dynamic formulation produces contingency plans whose implementation *depends* upon the outcome of future events. A contingency plan is composed of a course of action and contingent system state. Any particular course of action is only taken if the system state evolves according to what is laid out by the contingent system state. Hence, plans dynamically change with the state of the system.

Since the structure of our functional system includes multiple decision levels, a combination of both static and dynamic could be used if they are implemented on different levels. For instance, a static formulation might govern the behavior of CC while dynamic formulations might direct each of the sub-controllers. Below are some examples of each type of formulation along with their merits/drawbacks as they relate to our system. It is assumed the reader is familiar with the fields of Linear Programming (LP) and Dynamic Programming (DP).

4.1.1 Static Formulation

Composite Variable: Christopher Barth uses a composite variable formulation, which is a Mixed Integer Program (MIP) where the variables are heuristically generated, in an air operations scenario. His model (labeled CMVD) assigns aircraft and weapons to targets at specific times. CMVD includes routing and refueling decisions. A composite variable, by definition, reduces multiple decisions into one variable. In CMVD, the composite variables define which missions start at what times. The "mission" includes the target, type of aircraft tasked, bases that provide the aircraft, routes and assembly points. If every possible mission were included, the enormous number of variables would make CMVD intractable for all but the smallest cases. The main attractive feature of a composite variable formulation is that it has the potential of dramatically reducing the number of variables in the model. CMVD generates, based on heuristics and price-coordinated decomposition, all the mission variables for inclusion in the model. Price-coordinated decomposition proves useful here because the number of variables (missions) in the solution is small compared to the number of feasible missions.

Barth's model is static because it formulates a plan for the entire planning horizon that is not dependent on future world events. In other words, it does not include contingency actions for projected future events. Even if 1000 new targets are discovered in the course of the planning horizon, the model's decisions do not change. To compensate for this as well as other contingencies, Barth uses a rolling horizon in his planning algorithm. Even though the original plan has a horizon of perhaps 24 hours, the algorithm will re-plan after only, say, one hour. Therefore, all of the information gained in the first simulated hour can be accounted for in subsequent plans [4].

Robust Optimization: In certain problem domains, the solution to a MIP might be rendered meaningless if the data to the problem changes or turns out to be different from what was originally thought. The goal of Robust Optimization is to protect against uncertainty in the data. It involves the reformulation of a MIP (or an LP) into a more conservative version of the problem where feasibility is not affected by small data changes. For instance, suppose we have a MIP and the data is uncertain enough that there is a significant probability that a MIP solution will be infeasible. Using Robust

Optimization, the problem can be reformulated in a way that produces a slightly sub-optimal solution while driving down toward zero the probability of an infeasible solution.

An attractive feature of this method is that it allows the user to select the degree of conservatism desired. In other words, one can manipulate the "protection" parameter to achieve a desired level of probability that no constraints are violated. Also, it does not have the "curse of dimensionality" often found in stochastic optimization; a robust reformulation has, at most, twice as many constraints as the original problem. Finally, the robust solution has the potential to come very close to optimality. In one example, Bertsimas and Sim obtain a constraint violation probability of only 5.04×10^{-9} and still maintain a solution within 3.29% of optimality. To produce such results, however, the assumption is made that each uncertain data coefficient is independently and symmetrically distributed [5], [6]. At any rate, air and space operations involve much uncertainty and the new field of Robust Optimization could have the potential of ensuring feasible, if not completely optimal, solutions in this setting.

4.1.2 Dynamic Formulation

Stochastic Programming: Stochastic Programming (SP) breaks down the decisions of an LP or MIP into stages. An SP model has variables that represent current decisions as well as future, contingent decisions. Once the model has solved, these future decision variables (contingency variables) are the optimal decisions under a given future system state. As an example, suppose we have a problem covering two time periods. The system state in first time period is known. Now suppose there are four possible states that the system could be in at the second time period. After solving the SP, the optimal policy is this: we 1) implement the first-stage optimal decisions, 2) wait and see which state of the world transpires, and 3) implement whichever second-stage decisions correspond to that state. The objective function of an SP includes the contributions from first-stage decisions as well as the expected value of contributions from second-stage decisions. The constraints of an SP problem follow a "block-ladder" structure and can be solved by methods such as Bender's decomposition [7].

Robert Murphey models the Weapon Target Assignment (WTA) problem as an SP. The WTA problem seeks to optimally assign weapons to targets so as to maximize the total expected damage. Uncertainty involving the system state arises because the probability that a given weapon will destroy a target is less than 1. As a further complication, only a subset of targets is known at any given time. Murphey formulates the WTA problem as an integer, non-linear, two-stage SP. After decomposing the problem by stage, he provides an appropriate approximation algorithm [18].

This type of problem is computationally hard due to non-linearity, integrality, and the enormous number of constraints usually associated with SPs. Also, there is no explicit measure of how close Murphy's algorithm approximates the true solution. In addition, the WTA formulation looks only one time-step in the future and, for our scenario, would not take overarching, strategic goals into account. As a last note, stochastic programming in general assumes that future states of the world are independent of current decisions. This assumption does not hold for *ASOPP*.

Dynamic Assignment Problem using Dynamic Programming: Spivey and Powell examine the dynamic assignment problem (DAP) in [24]. DAP is the problem of dynamically assigning resources to tasks where both resources and tasks can randomly appear in the system. It is assumed that once a resource is assigned to a task, both the resource and task vanish from the system. Also, it is assumed that the world evolves in a Markovian manner – whatever happens in the next time-step is only dependent on the current state of the world. A DP formulation is proposed in [24]. Dynamic Programming often involves an enormous state space for realistic problems. However, there is an approximation algorithm provided in the paper that uses a linear estimation for the DP value function. This considerably shrinks the state space. One attractive feature of DAP is that it captures the value of holding on to a resource for future use. Spivey and Powell also provide a discussion on when it is appropriate to use myopic methods versus dynamic methods. Not surprisingly, for most cases in their research, dynamic methods outperform myopic methods (due to the value of estimating future data). However, in some cases, when enough advance information is available, the myopic methods do provide a better solution [24].

4.1.3 Combination of Both

LP/POMDP Hybrid Approach: Eric Zarybnisky, extending an approach by Yost [26], employs a combination of LP with Partially Observable Markov Decision Processes (POMDP) to solve the *Targeting Cycle Problem.* There are certain differences between the *Targeting Cycle Problem* used by Zarybnisky and the *Targeting Problem* used by Barth. The main difference is that Zarybnisky includes the ISR function where Barth does not. Zarybnisky explicitly models the finding of new targets, identifying targets, and confirming the destruction of previously attacked targets.

POMDPs are explored in the context of air operations due to their superior ability to model the probabilistic transitions inherent in battlefield scenarios. POMDPs model decision processes that have Markovian properties. They differ from regular MDPs in that the true state of the system after each action is taken is not known exactly.

In this hybrid method, a master LP is used to maximize objectives relating to each target type. The POMDPs are used to generate columns, or potential policies, for each target type. A policy with respect to a target is fundamentally a decision tree, where each branch represents a different action to take under certain observations. As with most column generation techniques, columns are generated and the LP is repeatedly solved until a master LP solution within a certain tolerance is reached [27].

4.2 Software Architecture and Constraints

The system architecture is shown in Figure 4-1. In the diagram, the five boxes represent functional components while the nine ellipses represent information flows. This figure is labeled the *Coordinated Dynamic Air and Space Operations Control System (CDASOCS). CDASOCS* is different from the functional diagram presented in Chapter 3 (*FCASOS* in Figure 3-1), primarily due to software constraints. In *FCASOS,* CC passes information to the sub-controllers, the sub-controllers give resource tasking to the SIM, and the SIM, at the end of each replan interval, updates CC.

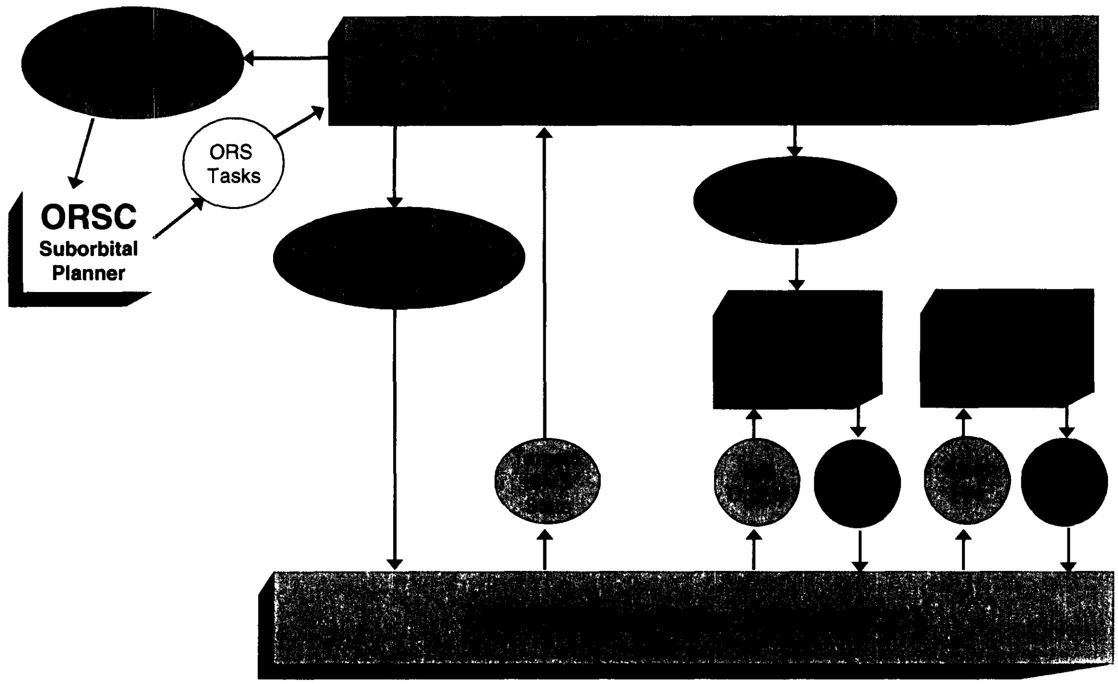

Figure 4-1: Coordinated Dynamic Air and Space Operations Control System (CDASOCS)

The actual software implementation, shown with *CDASOCS,* differs from this in the following ways: 1) The tasking from ORSC, instead of going directly to the SIM, is passed through CC first. This particular change is beneficial because CC can use ORSC tasking information to better coordinate with the other sub-controllers. 2) The AC sub-controller is no longer directly linked to CC. Instead, directions from CC to AC pass through the SIM. The reason has to do with the existing AC software, called IBMS (discussed in section 4.3). IBMS has its own information model, which is spread across a number of different files and includes a tightly-woven communications link to the SIM. Any input into IBMS must first go through its information model. Extracting the information model from IBMS and having CC send it input would be a massive undertaking and involve a reworking of much of the IBMS code structure.

Algorithms that are to be integrated into the *CDASOCS* system must satisfy the constraints presented by existing software. The following constraints have mainly to do with what each software program can accept as inputs and produce as outputs. For instance, in the cases of AC and SATC, data passed from CC must be of the type and in the form that they can accept.

1. **Lack of Sub-controller Feedback**
2. **Inability of SATC to Accept Time Sequencing Information**
3. **Inability of AC to Accept Target Specific Tasks**

Each constraint is discussed below and it is shown why their relaxation would yield more effective coordination.

Lack of Sub-controller Feedback: Currently, neither the AC nor the SATC program is configured to provide CC with feedback after producing a plan. As a consequence, CC does not find out AC and SATC resource tasking for a given planning interval until after that planning interval is over. Even then, CC can only infer sub-controller actions based on the updated resource and target state information. Feedback is a crucial part to effective coordination and this constraint significantly limits CC's ability to cause sub-controllers to work collaboratively. Consider the following: suppose AC is tasked to strike two targets and SATC is tasked to perform the post-strike BDA. It is assumed that the commander's intent values for the two targets are very close. Also, the desired action on each target is a strike followed by an immediate BDA. AC then determines it is only feasible to strike one of the two targets, so it chooses the one with slightly higher value. On the other hand, SATC determines the higher-value target is infeasible to ISR so it performs ISR only on the lower-value target. In this case the effect of collaborative decision making is lost – AC will strike the higher-value target while SATC will ISR the lower-value target. If the sub-controllers could have relayed tasking information to CC before implementation, this situation could have been remedied. Updated tasking might have been sent where AC and SATC are both tasked with the lower value target, causing a strike and BDA in close succession, and accomplishing the commander's objectives.

Inability of SATC to Accept Time Sequencing Information: The existing SATC system, named EPOS and discussed in section 4.2, is configured to accept a weighted target list as input. While CC can discriminate among which targets to send to EPOS and among the relative target valuation, CC cannot provide the specific times it wishes for each target to be viewed. In effect, this prevents CC from sequencing EPOS with other sub-controllers *inside* a planning interval. For example, in one planning interval, CC could not expect SATC to perform ISR on a target where the effectiveness of this ISR is time-dependent on another sub-controller's actions. This constraint does not completely prevent coordination though. Suppose a target was struck in the previous planning interval but not yet BDA'd. CC could then task EPOS to perform BDA in the *upcoming* interval and discourage the other sub-controllers from doing the same.

Inability of AC to Accept Target Specific Tasks: Similar to SATC, the existing AC system (IBMS) does not accept time-specific tasks. More than this, IBMS does not accept target-specific tasks either. Rather, input to AC comes in two tables. One table is a *commander's priority table* that lists relative target weight, but only according to target type and region. Each value in this table corresponds to the relative weighting of all targets of the same type and in the same region. The *commander's priority table* limits coordination because, under this framework, CC cannot influence AC to attack specific targets, just subsets of targets. Careful precaution must be taken to ensure AC and one of the other sub-controllers do not process the same target at the same time. The second is a *tolerable risk table,* given by aircraft type. This is a measure of the biggest acceptable probability of destruction for each aircraft type on any given mission.

We go into more depth about IBMS and EPOS next, in sections 4.2 and 4.3.

4.3 Existing Aircraft Planner (IBMS 1.0)

The pre-existing AC software, named the Integrated Battle Management System (IBMS) version 1.0, includes both a heuristic and an optimizing planner (see [2] for information on the decision structure used in IBMS). The heuristic planner is used in this thesis. The user may specify everything from number and type of aircraft and aircraft bases, to specific aircraft characteristics, such as average cruising speed.

As pointed out previously, IBMS has its own information model of the battlespace, which estimates the state of each target as well as the threat level in each battlespace region. The information model is updated at the beginning of each planning interval. Also, it can be made to operate with three different levels of initial target knowledge, including: 1) no initial knowledge, 2) nominal knowledge, and 3) perfect knowledge. IBMS is capable of tasking missions for multiple aircraft bases, aircraft types, weapon options, and against multiple target types. Specifically, IBMS assigns aircraft with missions, plans the packages to be used for each mission, makes weaponeering decisions for each strike objective, and determines the most effective routing for aircraft to ingress/egress the target area.

4.4 Existing Satellite Planner (EPOS 2.0)

Draper Laboratory's Earth Phenomena Observing System (EPOS) version 2.0 is used as the SATC component in *CDASOCS* [1]. EPOS optimally assigns satellites and their sensors in a given constellation to maximize the coverage of points of interest on earth. Coverage is defined as the accumulated viewing time of an object and is weighted by the object's relative importance. Integer programming, network optimization, and astrodynamics are used to generate sensor tasking plans. In addition to being able to specify the number, type, and orbital parameters of satellites as well as the number, type, and location of targets, the user defines 1) the field of view (FOV) on each sensor in an x and y directional plane and 2) the maximum sensor slew rate. The sensor slew rate is a measure of how fast a sensor can swivel from looking at one target to another. The slew rate greatly affects a satellite's ability to have multiple looks in one pass, where a *look* is defined as a single picture taken by a single satellite sensor. A sensor may, in some cases, be able to see multiple targets with one look if the field of view is big enough. Otherwise, the sensor must slew to see more than one target. It is assumed there is no cloud cover over the target area for the duration of this scenario. The satellite tasking produced by EPOS includes: 1) a list of targets, 2) the time window when each target is to be viewed, and 3) the specific satellite and sensor that is to view each target (See [1] for a complete description of EPOS).

4.5 Coordination Controller Algorithms

Nominal, heuristic, and optimal coordination algorithms are presented in this section. Each algorithm represents a different way of implementing the CC function of Chapter 3. The nominal version is formulated as the control, allowing other algorithms to be tested against it to quantify, in some sense, the value of sub-controller coordination. First, the indices, inputs, and decisions applicable to all three algorithms are listed.

Base CC Initialization

Indices

1. a – Aircraft index. $a \in \{1,...,A\}$
2. c – Controller index. $c \in \{AC, ORS, SAT\}$
3. h – Threat index. $h \in \{1,...,H\}$
4. i – Target index. $i \in \{1,...,I\}$
5. j – Aircraft type index. $j \in \{1,...,J\}$
6. l – Target type index. $l \in \{1,...,L\}$
7. p – Campaign phase index. $p \in \{1, 2\}$
8. t – Time period. $t \in \{1,...,T\}$

Inputs from Commander (Parameters)

1. V_i – Commander's Intent value for target i. $V_i \in [0, 1000]$

2. $tgtThreshold_l$ – Damage threshold required for a target of type l. $tgtThreshold_l \in [0,1]$. The commander specifies in this parameter the damage percent required for a target of a given type to be considered destroyed.

3. $ORSCostParam$ – ORSC resource cost parameter. $ORSCostParam \in [0,1]$. A measure of how costly it is to launch an LV. A value of 1 means there is no cost. A value of 0 means there is such a high cost that LVs will not launched.

4. $ORSvalueThreshold$ – Value threshold above which, targets will be assigned to ORSC. $ORSvalueThreshold \in [0, 1000]$. This is used to quantify the what type of targets get sent to ORS assets. A value near 0 means ORSC will go after virtually any target. A value near 1000 means only the highest-valued targets are considered.

5. $estCAVHits_i$ – Number of estimated CAV hits required to cause target i to exceed its damage threshold. $estCAVHits_i \in \{1,...,5\}$

6. $ObserveDecayParam$ – Time decay parameter for target observations. $ObserveDecayParam \in [0,1]$. This models the transient nature of battlefield ISR information. A value of 0 means the accuracy of any ISR observation does not degrade with time. A value of 1 means accuracy degrades rapidly and that targets must be observed at a fairly regular rate.

Inputs from SIM (IPB)

1. O_i – Last observation time of target i. $O_i \in \{unObserved,\ [0, current\_time]\}$. The target has either not been observed yet or has been observed at a time that is between the campaign start time and the current time.

2. S_i – Damage State of target i. $S_i \in \{0 - nothing,\ 1 - Damaged,\ 2 - Destroyed\}$. A three-state representation of target damage state is used.

3. R_i – Mission status of those missions planned against target i. $R_i \in \{0 - nothing,\ 1 - ACstrikeEnRoute,\ 2 - SOVstrikeEnRoute,\ 3 - strikeComplete,\ 4 - BDAcomplete\}$. This value tells CC one of the following: *0* – there has not been any strike missions planned yet, *1* – there is an aircraft en route for strike, *2* – there is an SOV en route for strike, *3* – a strike has occurred but not BDA, *4* –The target has been struck and BDA'd.

4. C_i – Target category of target i. $C_i \in \{1,..., J\}$.

Decisions (passed from CC to sub-controllers)

1. $TgtWeight_{ic}$ – Weight given to controller c for target i (0 if not assigned)
2. $newACrisk_j$ – New tolerable risk level for aircraft of type j.

Because the nominal and heuristic versions of CC are similar in structure, we present, in figure 4-2, a flowchart that applies to *both* the nominal and heuristic CC. The flowchart serves as a basic overview to the processes in these algorithms.

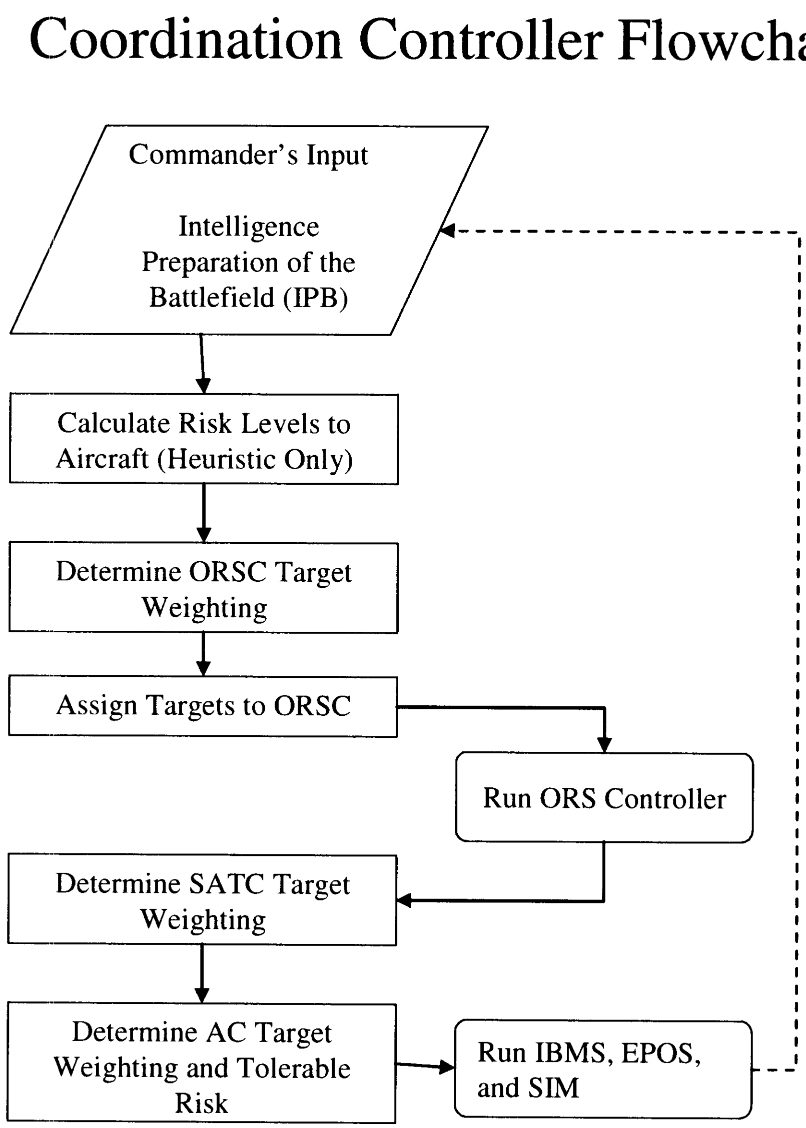

Figure 4-2: Coordination Controller Algorithmic Flowchart

4.5.1 Nominal Air and Space Operations Coordinator (*NASOC*)

The Nominal Air and Space Operations Coordinator (*NASOC*) represents a limited amount of coordination in the sense that sub-controllers: 1) are given the same IPB at the start of the campaign and 2) are updated with the same target state information at every replan. One characteristic that makes *NASOC* a realistic baseline coordinator is this: while sub-controllers are updated with the state of their own resources, they are not given the state of those resources that belong to other sub-controllers. Often in the real world, this type of communication breakdown contributes to the ineffective use of resources. In addition, *NASOC* does not accept feedback from sub-controllers (even if it could, this would only be applicable to ORSC). This is a greedy algorithm in the sense that CC tells sub-controllers to work independently to achieve their own objectives.

NASOC Algorithm

1. Determine ORSC target weighting

For all targets, the ORSC weighting is initially set to the commander's intent value. That is,

$$tgtWeight_{i,ORS} = V_i \tag{4.1}$$

Then, if the target has either been BDA'd and found to be destroyed or, has yet to be observed, the ORSC weight is set at zero:

$$tgtWeight_{i,ORS} = 0 \tag{4.2}$$

Note that the weight is not influenced in this algorithm by what other sub-controllers did or are capable of doing. Here, ORSC only has access to the IPB and only the results of ISR missions get put into IPB.

2. Assign targets to ORSC

We calculate $MaxNumAssign$, which is the upper limit to the number of possible target assignments to ORSC, as follows: First, we assign weights to the the amount of time remaining in the campaign (4.3) as well as the amount of remaining target value (4.4). Each of these is a number between 0 and 1. Both have decreasing marginal values related to the time remaining and value remaining, respectively.

$$timeWeight = \sqrt{\frac{campaignLength - currentTime}{campaignLength}} \tag{4.3}$$

$$valueWeight = \sqrt{\frac{remainingTgtValue}{beginningTgtValue}} \tag{4.4}$$

Then, we calculate the fraction and number of ORS assets to assign (4.5 and 4.6) using the $ORSCostParam$ as an additional weight. In this way, ORSC is encouraged to use more assets when: 1) the commander is less conservative, 2) there is a larger fraction of target value left on the battlefield, and 3) there is a larger fraction of time left in the campaign.

$$ORSAssignFraction = ORSCostParam * valueWeight * timeWeight \tag{4.5}$$

$$MaxNumAssign = ORSAssignFraction * numAvailableCAVs \tag{4.6}$$

Next, targets are assigned to ORSC in decreasing order of $tgtWeight_{i.ORS}$. Each target is assigned to however many CAVs the commander thinks is needed to destroy a target of that type, labeled $estCAVHits_i$. Targets are assigned until either the maximum number of CAVs ($MaxNumAssign$) is reached or until ORSC value is below the value threshold: $tgtWeight_{i.ORS} < ORSvalueThreshold$.

3. *Run ORS Controller (heuristic or optimal)*

a. *Assign CAVs to targets*
b. *Assign SOVs to CAVs*
c. *Assign launch times to SOVs*

4. Determine SATC target weighting

For all targets, the SATC weighting is initially set to the commander's intent value:

$$tgtWeight_{i.SAT} = V_i \tag{4.7}$$

Then, the target weight is multiplied by a time decay function. The decay function is calculated as follows: first, the fraction of the campaign length since the target was last viewed is determined:

$$timeSinceLastObserve = \frac{(currentTime - O_i)}{Campaignlength} \tag{4.8}$$

Note that $timeSinceLastObserve$ is in [0,1]. Now, at the beginning of the campaign, some targets are in the initial IPB as being observed, meaning the target was viewed before the campaign started, while others are listed as unobserved. For those listed as unobserved, the $timeSinceLastObserve$ variable is given a value of 1. Then, the decay factor is calculated:

$$decayFactor = ObserveDecayParam * (1 - timeSinceLastObserve) \tag{4.9}$$

In effect, the decay factor is closer to 1 for recently viewed targets and closer to 0 for targets that have not been viewed in a long amount of time. The SATC target weight is calculated as a decreasing function of this decay factor:

$$tgtWeight_{i,SAT} = tgtWeight_{i,SAT} * 1.5 * e^{-decayFactor} \tag{4.10}$$

From this equation, the SATC target weight could increase to 1.5 times the previous value, if the target has never been viewed. Alternatively, the target weight will decrease if recently viewed. The amount of decrease is dependent on the $ObserveDecayParameter$.

Next, the SATC target weight is set to zero if the target has already been BDA'd and found to be destroyed:

$$tgtWeight_{i,SAT} = 0 \tag{4.11}$$

Lastly, targets are assigned to SATC until max capacity is reached.

5. Determine AC target weighting and AC tolerable risk level

For all targets, the AC weighting is set to the commander's intent value:

$$tgtWeight_{i,AC} = V_i \tag{4.12}$$

If no ISR aircraft are left in the campaign, it is pointless to send out aircraft strike missions to any targets other than high-value targets. The destruction of a high-value target (WMD) could be beneficial, even without BDA, because only their destruction is sought. However, the reason for destroying other targets is not for their destruction, but for the effects of their destruction. For instance, destroying a SAM helps to gain air superiority and clears a pathway to high-value targets. But BDA of the target is required before the effect of their destruction is realized. Although SATC theoretically could do this BDA, this option is not taken into consideration because of the uncoordinated nature of this algorithm. SATC does not know what other sub-controllers can do or have done and the only way SATC would perform the BDA of an aircraft-struck target is by coincidence. Therefore, if no ISR aircraft remain, the values of all targets except WMD are set to zero:

$$tgtWeight_{i \notin WMD,AC} = 0 \tag{4.13}$$

The tolerable risk level for all aircraft is set to whatever the commander determines. To compare the aircraft attrition rate under different versions of CC, this number must remain constant across all algorithms:

$$newACrisk_j = TolerableRiskLevel_j \tag{4.14}$$

4.5.2 Heuristic Air and Space Operations Coordinator (*HASOC*)

The Heuristic Air and Space Operations Coordinator (*HASOC*) is the next level up from *NASOC* in terms of coordination. In *HASOC*, more resource information is shared among sub-controllers and decisions are made in more of a collaborative manner, taking into account the capability and state of sub-controllers. Algorithmically, the similarities and differences between *HASOC* and *NASOC* are this: They both determine the target weighting to pass to sub-controllers. However, *NASOC* uses *only* the commander's intent and IPB information to determine the target weighting; *HASOC* uses the commander's intent and IPB, along with the state and capability estimation of other sub-controllers' resources.

HASOC is constrained by certain time-sequencing limitations of the *CDASOCS* system. For instance, CC does not task sub-controllers with the specific times to take actions. While CC can use knowledge of sub-controller capability to better assign targets and target weights, CC cannot directly cause an ID-STRIKE-BDA sequence against a target using different sub-controllers.

There are two main additions to *NASOC* in this algorithm. The first is the use of two campaign phases. Phase one lasts for the first few hours of the campaign and is used mainly to clear a corridor through the high-threat areas so that AC can get to the high-value areas later. Second is the use of sub-controller estimated capabilities. CC attempts to gauge the capability of each sub-controller and use that information to make more effective target weighting and assignment decisions. In addition to the inputs given in the base CC initialization, the following is needed for *HASOC*:

Additional Inputs from Commander (Parameters)

1. $RiskParam_p$ – The fraction of ORSC strike value that should come from risk estimates versus commander's intent value for phase p.
$RiskParam_p \in [0,1]$

2. $TolerableRiskLevel_j$ – This number is a function of the maximum probability of attrition tolerated for aircraft of type j. $TolerableRiskLevel_j \in [0, 999]$. This parameter is used as a constraint on all aircraft of a similar type. It requires that any particular aircraft not go on a mission where the risk level is greater than the given parameter value.

3. $ISRdistWeight_i$ – A measure of how much to penalize SATC target weight if there are ISR resources close to the target. $ISRdistWeight_i \in \{1,\ldots,50\}$

Additional Inputs from SIM (IPB)

1. L_a – latitude/longitude location of aircraft a. $L_a \in [\{0,0\},\{180,180\}]$
2. $threatRadius_h$ – Lethal threat radius (in km) of threat h. $threatRadius_h \in [0,200]$
3. $threatLevel_h$ – Threat level of threat h. $threatLevel_h \in [0, 999]$. This is a measure of the likelihood that an aircraft will be shot down if in the threat radius of threat h.

***HASOC* Algorithm**

1. Calculate risk levels to Aircraft

An estimate is needed of how risky it is for AC to attack any given target. Risk, as it is used here, is one way of quantifying the likelihood that an aircraft will get shot down on a mission to attack a given target. If a target is extremely risky, CC might assign it to ORSC instead of AC to decrease the chance of aircraft attritions. Because there is just one aircraft base in the scenario, only the routes from that base to each target need to be considered. Also, only straight-line routes are used for the sake of simplicity and so as to not completely replicate the aircraft router.

For each target, the slope and y-intercept of the line from base to the target are calculated:

$$slope = \frac{(tgtLat - baseLat)}{(tgtLon - baseLon)} \tag{4.15}$$

$$yIntercept = baseLat - (slope * baseLon) \tag{4.16}$$

Then, the Euclidean distance from base to the target is determined:

$$distBaseTgt = \sqrt{(baseLon - tgtLon)^2 + (baseLat - tgtLat)^2} \tag{4.17}$$

Next, we loop over all other targets that: a) are not yet destroyed b) have no strike mission currently en route, and c) are listed as threats to aircraft. We refer to these targets as threats. For each of these threats, we need to determine if and how much of the straight-line from base to the target lies within the lethal threat radius of the threat. The intersection points (if any) between the line and the threat radius are calculated as follows. We substitute the equation of the line into the equation of the threat circle and solve for the coefficients to the quadratic equation:

$$A = 1 + slope^2 \tag{4.18}$$

$$B = (2 * slope * b) - (2 * circleLon) - (2 * slope * circleLat) \tag{4.19}$$

$$C = b^2 - (2b * circleLat) + circleLon^2 + circleLat^2 - circleRadius^2 \tag{4.20}$$

If the solution to the quadratic formula is not imaginary, then two intersection points of the circle and line exist, and are given by:

$$x1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \qquad y1 = slope * x1 + b \tag{4.21}$$

$$x2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \qquad y2 = slope * x2 + b \tag{4.22}$$

Now, it could be that the threat lies beyond the target, and thus is unimportant. To check for this, we find the distance from base to each of the intersection points and make sure they are less than the distance from base to the target (using Euclidean distance again). If both are greater than base-to-target distance, then the threat is beyond the target and nothing is done. If only one is greater, the target happens to be within the threat radius, so only the distance from the first intersection point to the target is counted. If neither is greater, the target is beyond the threat and the whole line lying within the threat circle should be counted. The counted distance within the threat circle is labeled $distRisk$. It is multiplied by $threatLevel_h$, which is a measure of the threat's lethality, and this is added to the target(not the threat)'s total risk level. Thus:

$$distRisk_i = threatLevel_h * countedDist \tag{4.23}$$

In addition to target risk levels, which represent the danger involved in getting to a target, $threatLevel_h$ also needs to be factored back in to the value of the threat itself. If a threat is on the straight-line path to multiple high-value targets, then ORSC could be used to destroy it and clear the way for future AC missions to the high-value target area. We weight the threat increase by a percentage of the intent value of the target:

$$threatAddedValue_j = threatAddedValue + \frac{distRisk_i * V}{1000} \tag{4.24}$$

2. Determine ORSC target weighting

For all targets, if there are no aircraft in the scenario, the ORSC target weighting is initially set to the commander's intent value:

$$tgtWeight_{i,ORS} = V_i \tag{4.25}$$

However, if there are aircraft, the ORSC weighting is initially set to a function of the commander's intent value, the target's risk level ($distRisk_i$), and the target's added threat value if it is also a threat, labeled $threatAddedValue_j$:

$$tgtWeight_{i,ORS} = (1 - RiskParam_p) * V_i + (0.5 * RiskParam * distRisk_i) + (0.5 * RiskParam * threatAddedValue_j) \tag{4.26}$$

Then, there are three cases where the ORSC weight will be set to zero: 1) if the target has already been BDA'd and found to be destroyed; 2) the target has yet to be viewed by an ISR resource; or 3) a previous ORSC or AC strike mission was sent against the target and, although not BDA'd yet, it is likely the target is destroyed. If any of these is the case then the ORSC weight is set to zero:

$$tgtWeight_{i,ORS} = 0 \tag{4.27}$$

3. Assign targets to ORSC

First, $MaxNumAssign$ is calculated using equations 4.3-4.6 from *NASOC*. Now, it is the goal of *HASOC* to use ORSC to create a corridor through the SAMs so that aircraft may get to the higher-value targets. The corridor is defined by specific lat/lon coordinates and is wide enough so that aircraft could pass through the middle without risk of attrition. However, if a SAM in the corridor has yet to be viewed yet by an ISR resource, ORSC is barred from striking it. To ensure ORSC has enough resources to create a corridor, $MaxNumAssign$ is adjusted by $numCAVsNeededForCorridor$, which is the number of CAVs needed to strike all SAMs in the corridor that have yet to be viewed.

$$MaxNumAssign = MaxNumAssign - numCAVsNeededForCorridor \tag{4.28}$$

Next, targets are assigned to ORSC in decreasing order of $tgtWeight_{i.ORS}$. Each target is assigned to however many CAVs the commander thinks is needed to destroy a target of that type, labeled $estCAVHits_i$. However, there is a difference between this calculation and the one in *NASOC*. Here, if the target has already been partially damaged by AC, then the number of CAVs needed to destroy that target is decreased by 50%. After this, targets are once again assigned to ORSC until either the maximum number of CAVs ($MaxNumAssign$) is reached or until ORSC value is below the value threshold: $tgtWeight_{i.ORS} < ORSvalueThreshold$.

4. ***Run ORS Controller (heuristic or optimal)***

a. *Assign CAVs to targets*
b. *Assign SOVs to CAVs*
c. *Assign launch times to SOVs*

5. **Determine SATC target weighting**

In this section, a max SATC weight of 2000 is used. This is twice the max intent value and allows for scaling the intent value up or down. We distinguish among four different importance levels for the EPOS weight. The highest importance is given to SAMs inside the corridor that have yet to be viewed by an ISR resource. Viewing these SAMs quickly is crucial to the progress of the campaign. Once viewed, ORSC can strike them, the aircraft can go through the corridor, and the campaign can get under way. The second highest importance is given to those targets that have just been struck by ORSC or AC and require BDA. The third highest is given to targets needing pre-strike ID where strike missions are currently planned against the target. Any targets that do not fit into the previous three categories fall into the last level of importance. For all targets, the SATC weighting is initially set to the commander's intent value:

$$tgtWeight_{i.SAT} = V_i \tag{4.29}$$

The commander's intent values represent the fourth level of importance and range in value from 0 to 1000. If the target is assigned to ORSC and there is an ORSC strike mission planned against this target, the SATC target weight is assigned a value such that it competes with the highest-valued targets from above. This is the third level of importance. In the equation below, l is the target type index for target i, where targets that are higher-valued have a lower type index.

$$tgtWeight_{i.SAT} = 1000 - 50 * l \tag{4.30}$$

The target weight is then multiplied by the same time decay factor that was used in *NASOC*, given by equations 4.8-4.10. The time decay factor is used at this point in the section for a reason. Targets with the highest two levels of importance (given next) are not intended to be influenced by the time decay factor. Thus, calculating the time decay factor now allows us to then separately calculate a new weight if the target is of the highest importance.

If the target has recently been struck by an ORS or AC asset and now requires BDA, it is given the second highest level of importance and given a value between 1000 and 1500:

$$tgtWeight_i = 1500 - 50 * l \tag{4.31}$$

If the target is a SAM in the corridor region, it is given the highest importance:

$$tgtWeight_i = 2000 \tag{4.32}$$

Now that target weight according to the four levels of importance has been assigned, we factor in an estimate of the AC ability to ISR target. The straight-line Euclidean distance from every ISR resource to the target is calculated, labeled L_a and the minimum of these is found, called $MinISRdist$. The closer an ISR resource is to a target, the more likely it is this resource will be able to quickly perform ISR of the target. Therefore, the SATC target weight is penalized (up to a maximum of 350) for having a close ISR resource, and increased for having the closest ISR resource extremely far away:

$$tgtWeight_i = tgtWeight_i - Max\left(350, \frac{ISRdistWeight}{MinISRdist}\right) \tag{4.33}$$

Next, the SATC target weight is set to zero (equation 4.11 from *NASOC*) if the target has already been BDA'd and found to be destroyed. Targets are assigned to SATC until max capacity is reached.

6. Determine AC target weighting and AC tolerable risk level

Equation 4.12 from *NASOC* is used to assign initial AC target weighting. Equation 4.13 is also used, but under slightly different circumstances. Because this is a coordinated algorithm, SATC is capable of doing BDA of an aircraft-struck target. Thus, equation 4.14 (which sets AC weight to zero) is only used if there are no ISR aircraft left *and* there are no satellites being used in the scenario.

Equation 4.14 from *NASOC* is used to set the tolerable risk level to whatever the commander determines for each aircraft type. Although this number must remain constant across different algorithms for testing purposes, it is temporarily modified for brief intervals in order to effect better coordination. Since CC does not give AC target-specific weights (only target-type-specific) it could be the case that ORSC and AC task a strike mission to the same target.

Also, AC could send a strike mission to a target that ORSC struck previously but has not been BDAd. This case can occur either at the point when ORSC missions are tasked or immediately after this. The following heuristic is given to mitigate times when greater than four targets could potentially be struck by both AC and ORSC:

If LVs are en route to strike at least four targets, or if ORSC has struck more than four targets that have not been BDAd:

$$newACrisk_{j \in Strike} = 0 \tag{4.34}$$

Else,

$$newACrisk_j = TolerableRiskLevel_j \tag{4.35}$$

4.5.3 Optimizing Air and Space Operations Coordinator (*OASOC*)

The Optimizing Air and Space Operations Coordinator (*OASOC*) represents the next step up from *HASOC* in terms of sub-controller coordination. The objective function, as outlined in Chapter 3, is a linear combination of expected target damage, expected attrition and operating cost, and expected time to completion. The multipliers of the linear combination are determined by the commander and reflect the overall importance of each objective function component. Instead of assigning targets to sub-controllers, this formulation incorporates the idea of assigning mission options. This is a composite variable approach in the sense that each mission option incorporates multiple decisions. Because the list of possible mission options is incredibly large, heuristics need to be used to generate mission options and to ensure the solvability of the formulation.

Mission Options: It is assumed sub-controllers are capable of performing some combination three possible mission functions, including: 1) identifying a target (ID), 2) striking a target (STRK), and 3) assessing the damage of a target (BDA). A mission option, then, is defined as a tasking of at least one sub-controller to perform a series of mission functions at specific times. A mission option coupled with a target assignment and a start time produces sub-controller tasking.

The following restrictions are imposed upon mission options: 1) No more than one of each type of function may be included in the option; 2) A BDA may not precede a STRK and neither a STRK nor a BDA may precede an ID; 3) The timing of tasked functions must be properly spaced out according to margins of error on sub-controllers.

For instance, suppose an ORSC strike mission has a strike window of 30 minutes and a SATC BDA mission has an ISR window of 30 minutes. It would not make sense to task a SATC BDA of a target within 30 minutes of an ORSC strike. The following are two examples of mission options:

Mission Option 1:

1) SATC+ID+0000 2) AC+STRK+1800 3) AC+BDA+3600

Mission Option 2:

1) ORSC+STRK+5400 2) SATC+BDA+7200

The format for mission options is: <<*(Sub-controller)+(function)+(time after start of mission option to execute)*>>. So in the first mission option, AC is to perform a strike 1800 seconds (30 minutes) after the start of the option. In the formulation below, mission options are appended with a single target number and a start time to produce CC assignments. Below, we show an example of a CC assignment for target #32 to start at time 7200.

CC Assignment 1:

1) ORSC+STRK+5400+#32+7200 2) SATC+BDA+7200+#32+7200

Here, ORSC should strike target #32 at time 12600. Then, SATC should BDA the target at time 14400.

Plan Horizon Phase: The plan horizon is partitioned into phases according to fixed time intervals. For instance, a plan horizon that is 7200 seconds long could have three phases where the first phase starts at 0000 seconds, the second phase starts at 3600 seconds, and the third phase starts at 5400 seconds. The purpose of the phasing is to ensure that each sub-controller does not get overloaded with tasks during any short interval of time. For each sub-controller and each plan horizon phase, the number of assignable mission options is constrained. The constraint is not for the start time of the mission option but rather for the time that the sub-controller would be busy under that mission option. The following is an Integer Programming (IP) formulation for *OASOC* with a linear objective function and linear constraints.

OASOC Formulation

Indices

1) t – Time period. $t \in \{1,...,T\}$
2) i – Target index. $i \in \{1,...,I\}$
3) m – Mission option index. $m \in \{1,...,M\}$
4) ψ – Sub-controller index. $\psi \in \{1,...,\Psi\}$
5) θ_t – Plan horizon phase at time t. $\theta_t \in \{1,...,\Theta\}$

Input Data

1) V_{mit} – Value of mission m against target i commencing in time t
2) C_{mit} – Cost of mission m against target i commencing in time t
3) $P(k_{mit})$ – Probability that mission m, starting in time t, will destroy target i
4) $S_{\psi\theta_t}$ – Set of mission options involving controller ψ in plan horizon phase θ
5) $N_{\psi\theta_t}$ – Assignment Capacity of sub-controller ψ in plan horizon phase θ

Decision Variables:

1) x_{mit} – {1 if mission option m is scheduled to commence against target i in time t, 0 otherwise}

Objective Function

$$MAX \sum_m \sum_i \sum_t \left[\left(V_{mit} * P(k_{mit}) - \underset{AC\ attritions}{E} [C_{mit}] \right) * x_{mit} \right] \qquad (4.36)$$

Constraints

$$\sum_{m}\sum_{t} x_{mit} \leq 1 \quad \forall i \tag{4.37}$$

$$\sum_{t}\sum_{m\in S_{\psi\theta_t}}\sum_{i} x_{mit} \leq \sum_{\theta_t} N_{\psi\theta_t} \quad \forall \psi \tag{4.38}$$

$$\sum_{t}\sum_{m\in S_{\psi\theta_t}}\sum_{i} x_{mit} \leq N_{\psi\theta_t} \quad \forall \psi,\theta_t \tag{4.39}$$

$$x_{mit} \in \{0,1\} \quad \forall m,i,t \tag{4.40}$$

Objective Function: For any mission, target, and time combination, there is an expected value and an expected cost. The expected value is simply the commander's intent value for that target times the probability that the mission will succeed in in it's purpose (ID, strike, BDA, or a combination of these). The expected cost is the cost of attrition times the expected number of attritions.

Constraints: Constraint set 4.37 requires that no more than one mission may be assigned per target. Constraint set 4.38 is a capacity restriction that limits each controller to not exceed their supply. Constraint set 4.39 is similar and confines controller assignments based on supply within each of the plan horizon phases. Lastly, constraint set 4.40 eliminates the possibility of fractional missions.

4.6 ORSC Algorithms

Heuristic and optimal algorithms for ORSC are presented in this section. The ORSC algorithms are used to assign EVs to strike targets, LVs to carry EVs, and launch times for tasked LVs. In the current implementation, ORSC is not used to launch microsats and this is one potential area of future work. Below, we give the indices and inputs common to both algorithms:

Base ORSC Initialization

Indices

1) c – CAV index. $c \in \{1,\ldots,C\}$

2) h – Threat index. $h \in \{1,...,H\}$
3) i – Target index. $i \in \{1,...,I\}$
4) l – Target type index. $l \in \{1,..., L\}$
5) p – Campaign phase index. $p \in \{1, 2\}$
6) s – SOV index. $s \in \{1,...,S\}$
7) t – Time period. $t \in \{1,...,T\}$

Parameters

1) $requestedP(dest_l)$ -- Requested Probability of destruction for a target of type l. $requestedP(dest_l) \in [0,1]$

Inputs from CC

1) W_i – Weight of target i given by CC. $W_i \in [0, 1000]$

Inputs from SIM (IPB)

1) $ExpectedP(dest_{\text{ln}})$ – Expected Probability of destruction for a target of type l using n CAVs (where n is constant for a given l. $ExpectedP(dest_{\text{ln}}) \in [0,1]$

2) S_i – Damage State of target i.
$S_i \in \{0-nothing, 1-Damaged, 2-Destroyed\}$. A three-state representation of target damage state is used.

4.6.1 Heuristic ORS Controller (*HORSC*)

The heuristic version of ORSC (*HORSC*) has three goals. First, for each target, *HORSC* must assign enough EVs to that target to achieve a certain probability of destruction (given by CC). Secondly, it tries to achieve the greatest weighted expected target damage for the whole planning horizon by assigning high-value targets first. This is a greedy heuristic and may not lead to the highest potential target damage if the high-value targets require significantly more EVs to be destroyed. Lastly, *HORSC* must stay within its resource conservation goals, which depend on the state of the battle and a commander's conservation factor. As discussed previously, there is no time-sequencing capability in the implemented version of CC. ORSC does not have to worry about when to strike a target other than using the simple rule that the quicker the strike, the better.

HORSC Algorithm

1. Determine how many EVs to use on each target

For all targets, we calculate $numEVrequired_i$, which is the number of EVs that will achieve the requested probability of destruction for that target. The procedure to do this is as follows. We start with $ExpectedP(dest_{\text{In}})$, which is the expected probability of destruction given n EVs. Logarithmic interpolation is then used to determine the number of EVs that will produce a probability of destruction closest to $requestedP(dest_l)$. An example is provided to illustrate: Suppose the requested probability of destruction on a target of type 1 is 0.9. Also, it is given that $ExpectedP(dest_{12})$ = 0.75. This means that 2 EVs have an expected probability of destruction of 0.75 for a target of type 1. Modeling this logarithmically yields the following equation, where M is the multiplier to be used in the probability of destruction graph:

$$1-e^{-2M}=0.75 \tag{4.41}$$

and solving for M, we get:

$$M=\frac{-\ln(0.25)}{2}\approx 0.693 \tag{4.42}$$

The resulting probability of destruction graph is shown in Figure 4-3:

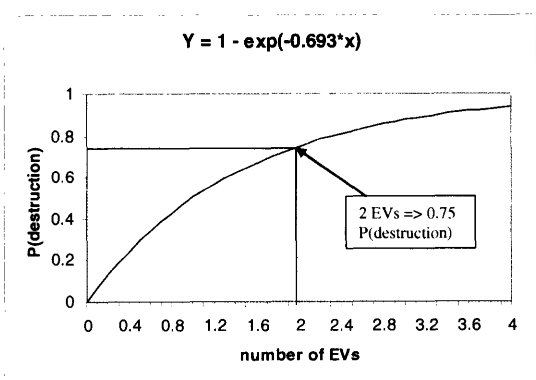

Figure 4-3: P(destruction) graph

Finally, we find the smallest number of EVs that yield a probability of destruction greater than or equal to 0.9 by plugging in successively higher numbers into the logarithmic equation. We see that 1 EV yields a P(destruction) of 0.5, 2 EVs yield 0.75, 3 EVs yield 0.875, and 4 EVs yield 0.9375. Therefore, to get a P(destruction) of at least 0.9, we need $numEVrequired_i = 4$.

Lastly, if $S_i = 1$, meaning the target is already damaged, then $numEVrequired_i$ is decreased by 50%.

2. Decide how many LVs to use and assign LV launch times

In the CC algorithm, an ORSC cost parameter was specified to give a measure of how costly it is to launch an LV. If it is very costly compared to expected target damage, not many LVs will be launched. Apart from this parameter, ORSC might have another reason for conserving resources. If it is expected that a large number of high-value or high-threat targets are going to appear before an LV could launch and get back to base, ORSC might want to conserve some LVs to be prepared for this. However, in the current scenario, all targets are known from the beginning. Also, ORSC has no way of projecting forward to determine how many targets it might get passed from CC at future replan intervals. Therefore, ORSC will always launch an LV that is available as long as 1) the cost of the launch (determined by a function of the ORSC cost parameter) is less than the expected value gained from the launch, and 2) there is at least one target left to attack. Also, since it is assumed that the quicker the strike, the better, all LV launch times are set to the first available launch time.

3. Assign each EV a target

A greedy heuristic is used here where the highest-value targets are assigned first. EVs are treated as indistinguishable, even when on different LVs. We start with the first EV and assign it to the target with the highest value. We continue assigning EVs to that target until either $numEVrequired_i$ is reached or until we run out of EVs. In the former case, the next highest-value target is assigned, and so on.

4.6.2 Optimizing ORS Controller (*OORSC*)

In the optimizing version of ORSC (*OORSC*), more realism is added by assuming that each LV has a probability of launch failure. With this assumption, two events must happen for an EV to impact a target. First, the LV must successfully launch. Second, the EV must successfully release from the LV, enter the earth's atmosphere, and glide to the location of the target. Whereas in the previous algorithm, the probability of these events happening was combined into one probability of success ($ExpectedP(dest_{\text{In}})$), they are now modeled separately.

The main effect of this assumption is that now the practice of cross-targeting – that is, attacking a target with EVs that come from *different* LVs – will yield a higher probability of destruction than using EVs from the *same* LV. Define L_l as the event that LV l launches successfully. Define K_{ie} as the event that EV e impacts and kills target i. Suppose the probability of a successful launch, or $P(L_l)$, equals 0.9 for all LVs. Suppose the probability of an EV hitting and killing a particular target given a successful launch, or $P(K_{ie}|L_l)$, equals 0.8 for all targets and EVs. Suppose EV1 and EV2 from the *same* LV (LV1) attack target #1. Then, the probability that at least one EV will hit the target is simply:

$$P\left(L_1 \cap \left(K_{11} \cup K_{21} | L_1\right)\right) = P(L_1) P\left(K_{11} \cup K_{21} | L_1\right) \tag{4.43}$$

The equality is based on an assumption of conditional independence. Then, assuming the independence of actions of different EVs after an LV launch, we obtain:

$$P(L_1) P\left(K_{11} \cup K_{21} | L_1\right) = P(L_1)\left(P(K_{11}|L_1) + P(K_{21}|L_1) - P(K_{11}|L_1) P(K_{21}|L_1)\right) \tag{4.44}$$

$$= (0.9)\left(0.8 + 0.8 - (0.8)(0.8)\right) = \mathbf{0.864} \tag{4.45}$$

This result is, of course, higher than if there were just a single EV (**0.72** in that case). Now suppose EV1 and EV2 came from *different* LVs. The probability that at least one EV will hit the target then becomes the probability of a launch and hit from LV1/EV1 or a launch and hit from LV2/EV2. This is written as:

$$P\left(\left(L_1 \cap K_{11}\right) \cup \left(L_2 \cap K_{21}\right)\right) = P\left(L_1 \cap K_{21}\right) + P\left(L_2 \cap K_{21}\right) - P\left(L_1 \cap K_{11} \cap L_2 \cap K_{21}\right) \tag{4.46}$$

$$= P(L_1) P(K_{11}|L_1) + P(L_2) P(K_{21}|L_2) - P(L_1) P(K_{11}|L_1) P(L_2) P(K_{21}|L_2) \tag{4.47}$$

$$= (0.9)(0.8) + (0.9)(0.8) - (0.9)^2(0.8)^2 = \mathbf{0.9216} \tag{4.48}$$

Therefore, it is evident that, under our assumptions, a higher probability of destruction is achievable by cross-targeting.

The *OORSC* formulation below is an IP with a non-linear objective function and linear constraints. The formulation itself is adapted from [25], although the idea of a cross-targeting, non-linear assignment problem has been around for some time. Auction algorithms were used in [25] to solve the IP. The design of the formulation is constructed so that, at each replan, we maximize the weighted expected target damage over the course of the next planning interval.

Before presenting the formulation, we need to extend the cross-targeting development to an expression that works for any number of EVs and LVs. Here, the probability that at least one EV hits the target is considered not in terms of unions of sets, but rather as one minus the probability that no EVs hit the target. Given E EVs, this is just:

$$1 - \prod_{e=1}^{E}\left(1 - P(K_{ie} | L_l)\right) \tag{4.49}$$

Next, we use equation 4.49 to find the probability of at least one successful mission, where a successful mission is defined as at least one LV (out of Λ total) launching *and* at least one EV hitting the target. This is one minus the probability of no successful missions and is given by:

$$1 - \prod_{l=1}^{\Lambda}\left(1 - P(L_l)\left[1 - \prod_{e=1}^{E}\left(1 - P(K_{ie} | L_l)\right)\right]\right) \tag{4.50}$$

This is used as an expression for the probability of destruction on any particular target, given Λ LVs and E EVs.

OORSC Formulation

Indices

1) i – Target index. $i \in \{1,\ldots,I\}$
2) e – EV index. $e \in \{0,\ldots,E\}$
3) l – LV index. $l \in \{0,..,\Lambda\}$

Input Data

1) $P(L_l)$ – Probability of successful launch of LV l
2) $P(K_{ie}|L_l)$ – Probability of destruction of target i from EV e given that LV l successfully launched
3) $NumEV$ – Number of EVs that can fit on an LV
4) W_i – Weight of target i given by CC. $W_i \in [0,1000]$

Decision Variables:

1) x_{iel} – {1 if target i is assigned to EV e on LV l, 0 otherwise}

Objective Function

$$MAX \sum_{i=1}^{I}\left[W_i\left(1-\prod_{l=1}^{\Lambda}\left[1-P(L_l)\left(\prod_{e=1}^{E}\left[1-P(K_{ie}|L_l)\right]x_{iet}\right)\right]\right)\right] \tag{4.51}$$

Constraints

$$\sum_{e=1}^{E}\sum_{i=1}^{I}x_{iel} \leq NumEV \ \forall l \tag{4.52}$$

$$\sum_{i=1}^{I}x_{iel} \leq 1 \quad \forall e,l \tag{4.53}$$

$$x_{iel} \in \{0,1\} \quad \forall i,e,l \tag{4.54}$$

Objective Function: Each target is given a weight, which reflects that target's relative importance to CC. In the objective function, this weight is multiplied by the total probability of destruction under any LV/EV combination. Thus, the purpose of the objective function is to maximize the weighted expected target damage. Once solved, the resulting values of the decision variables will represent the LV/EV to target combination that produces the maximum objective function value.

Constraints: Constraint set 4.52 limits the number of EVs that may be assigned from each LV to no more than what the LV is able to carry. Constraint set 4.53 requires that each EV be assigned to no more than one target. Lastly, Constraint set 4.54 defines the decision variables as binary variables.

[This Page Intentionally Left Blank]

5 Testing and Analysis

The purpose of this chapter is to gain additional insight into the air and space coordination problem through the testing and evaluation of CC and ORSC algorithms on an implemented *CDASOCS* system. The primary goals are to: 1) quantify the benefit of adding space operations to air operations, and 2) quantify the additional benefit of coordinating among resources. To give the reader an understanding of the software architecture used, details about the software structural integration are provided. The specific scenarios used for testing are then presented, including the base, or control scenario. Characteristics such as number and type of resources are incorporated into the scenario descriptions. Finally, a set of experiments are run to test certain hypotheses and to determine the effects of perturbations on the system and their results are given.

5.1 Software Integration

The integration of *CDASOCS* is complex. It involves four separate pieces of software running on three different computers that are connected over a local area network (LAN).

The different pieces of software include: 1) the battlespace SIM, 2) the AC program (IBMS), 3) the SATC program (EPOS) and 4) the CC/ORSC program written for this thesis.

Information is passed from one program to another by way of text files and "lurker" programs. The lurker programs are used to detect when new versions of text files have been written. To illustrate the process, consider the point in a replan interval when CC must produce new tasking to send to the SIM. The sequence of events for this to happen is as follows. First, CC and the SIM's lurker programs are already running (and has been since the beginning of the campaign). The SIM, once done simulating the previous interval (or preparing IPB for the first interval), writes updated IPB information to a text file on the LAN and closes the file. Meanwhile, CC's lurker program detects that the text file was just updated. The lurker then triggers CC to run the coordination algorithm, run the ORSC algorithm, and write the updated tasking in a second text file for the SIM. The SIM's lurker program detects that this text file was just updated. This alerts the SIM to read the updated tasking information, close the file, turn off its lurker, and start the next sequence. The whole process is now complete. The process just described is shown in Figure 5-1.

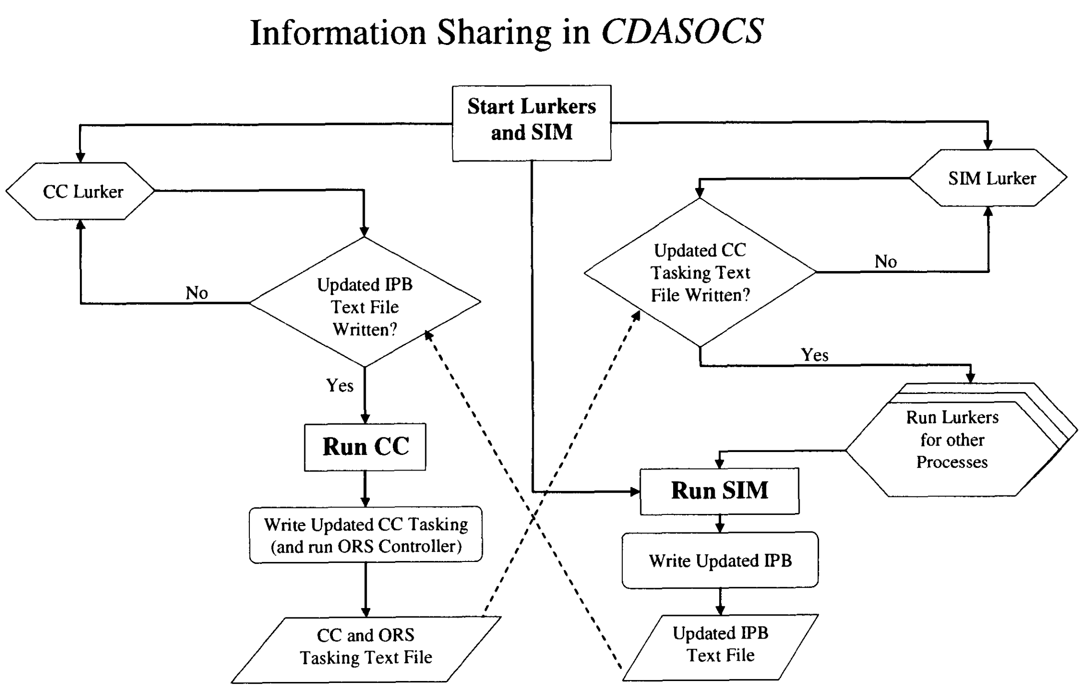

Figure 5-1: Illustration of Information Sharing in CDASOCS

Similar sequences to this one are used for every software piece that is connected.

CDASOCS runs on three different computers, and components are written in three different computer languages. The CC/ORSC program and EPOS run on the same computer – a Pentium 4 with 1.5GHz, 256MB of RAM, and Windows XP as the operating system. CC/ORSC is coded in JAVA while EPOS is coded in C++. Both use the optimization software, XPress-MP. The SIM runs on a 450MHz PowerPC G4 Macintosh with 256 MB of SDRAM. The operating system is OS X. The SIM is coded in Fortran. Lastly, IBMS runs on a 2GHz Pentium 4 with 1 GB of RAM and Linux as the operating system. IBMS is coded in C++.

5.2 Scenarios

In order to run experiments and gather meaningful data, there must be a somewhat-realistic scenario. This section discusses the major features of the scenario including geography, the target region, and all resource types. In addition to the major features, there are also a number of parameters in each component of *CDASOCS* that can be used to change how the system behaves. We list, for each component, those parameters that stay constant across all experiments as well as those allowed to change. Also, a base scenario is provided to serve as a means for initial analysis and as a control for the experiments.

The SIM has a built-in geodetic world map that can be used to model entities on any part of the globe. The aggressor state in the scenario is on the fictionalized Indonesian territory of Irian Jaya, the western portion of New Guinea. The target area occupies approximately 40,000km^2. There is one friendly air base set in the fictionalized Papua New Guinea. The base is approximately 350km to the east of the eastern edge of the battlefield in friendly territory. The ORSC-controlled LVs are stationed at Vandenberg Air Force Base in California. The scenario geography is shown in Figure 5-2.

Newtonian physics is used to model all LV and aircraft motion. Additionally, the satellites used by EPOS are in LEO at the start of the campaign. A campaign is defined as the predetermined length of simulated time that battlespace operations are to be planned for and executed. An orbital propagator in the SIM is used to model satellite motion.

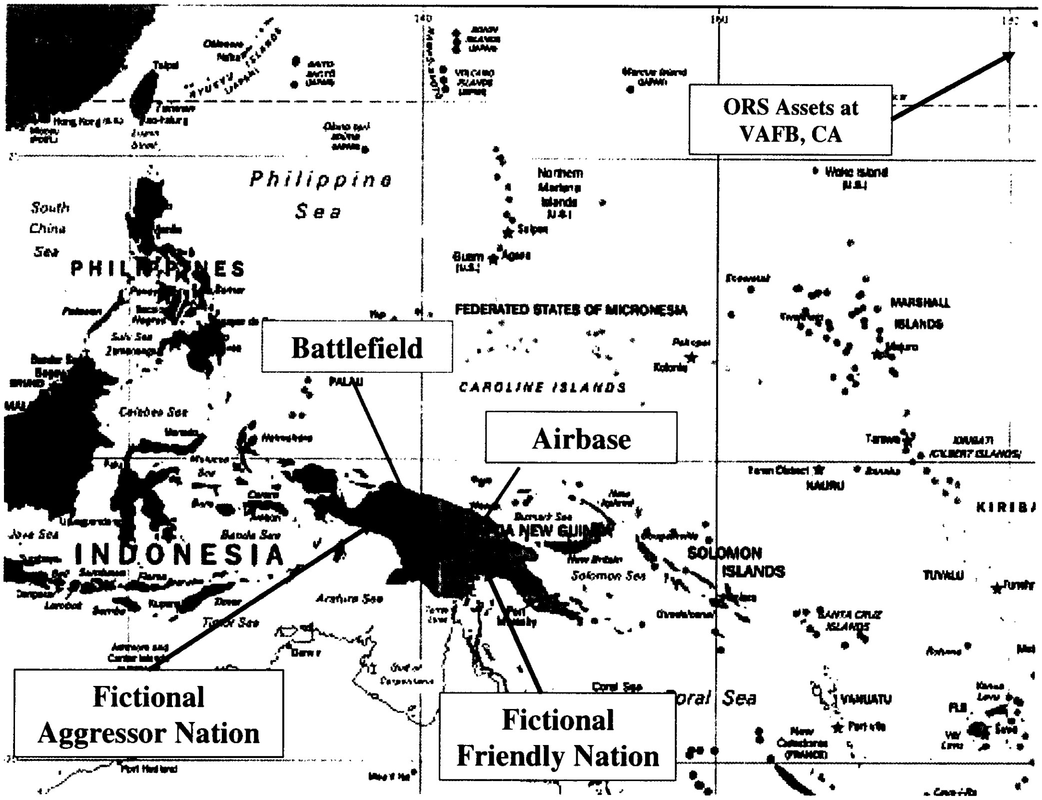

Figure 5-2: Scenario Geography

On the battlefield, there are four possible target types, all of which are considered stationary: 1) WMD targets are the highest-value targets and represent locations where the enemy is manufacturing WMD weapons. 2) C3CP targets are command and control facilities and are second in value only to WMD. 3) SAM targets include surface-to-air missile sites and are the only targets that pose a threat to aircraft. Their destruction is necessary to provide a clear pathway to the higher-value targets and to gain air-superiority over the battlefield. 4) Fixed targets are other targets of interest and encompass all target types other than the first three mentioned.

There are five different types of aircraft in the scenario. Two of these are ISR aircraft – one used for wide-area searches, one for specific target ID and BDA. Both types of ISR aircraft are modeled as Unmanned Aerial Vehicles (UAVs) that remain over the battlefield for the duration of the campaign. It is assumed they have adequate fuel provided. The other three types of aircraft are stationed at the air base and include two strike/fighter types and one jammer type. These aircraft have limited range and munitions. Each of the strike/fighter aircraft has a different weapons payload capability that governs its effectiveness against different target types. The jammer aircraft is used to block the radar used by SAM sites. Thus, if a jammer is in the vicinity of a SAM site, the probability of attrition for all aircraft in the area decreases. The specific speed and performance characteristics of all aircraft are modeled after the current inventory of US military aircraft.

5.2.1 Fixed Scenario Attributes

There are a number of parameters in each functional component that define certain functional attributes or behavior characteristics and which, for our purposes, will remain fixed across all experiments. We list each of these according to functional component and provide brief descriptions:

- **CC Fixed Parameters:**
    1. **tgtDamageThreshold** – The threshold over which a target is considered by the commander to be destroyed. For WMD and SAMs, it is set to 0.75. For C3CP and Fixed_targets it is set to 0.5.
    2. **ORSvalueThreshold** – Targets can only be assigned to ORSC if the calculated ORSC weight is above this value, which is set to 100.
    3. **CommandersIntent** – The commander's target importance weighting, given by target type and region:

Region	NW_Zone	NE_Zone	SE_Zone	SW_Zone
WMD	1000	1000	1000	1000
C3CP	500	500	500	500
Fixed_target	100	100	100	100
Fixed_SAM	300	300	300	300

4. **ACTolerableRisk** – The risk level that the commander is willing to accept for each aircraft type. Set to its max value for the purpose of comparing aircraft attrition rates.
5. **ISRAircraftDistWeight** – Measure of how much the distance from the target to the nearest ISR aircraft influences the satellite target weighting. A higher value means that the locations of ISR aircraft have a larger effect on the satellite-target assignment. For instance, a high *ISRAircraftDistWeight* means that targets with no ISR aircraft nearby are more likely to be tasked to satellites than targets with ISR aircraft in the area. After repeated testing, we set the value to 20.

- **EPOS Fixed Parameters:**
  1. **Plan Horizon** – Length of time for which EPOS plans when tasked. We set this to 0.75 hours, which is 0.25 hours longer than the replan interval. The replan interval is the length of time from the production of one plan to the production of the next plan. In our case, the replan interval starts when all sub-controllers have passed their tasking to the SIM, and ends when simulation is suspended so that a new plan may be generated.
  2. **Orbit Type** – Satellites are in circular LEO (altitude of 450km). There are 12 different orbital planes with the same inclination but different values for the Right Ascension of the Ascending Node (RAAN). The number of satellites in each plane depends on the scenario but the spacing between satellites is symmetric.
  3. **Max Target Capacity** – The maximum number of targets that can be assigned to EPOS. Repeated testing shows that EPOS never tasks more than 6 or 7 targets. Therefore, we set the value of this parameter to 20 to keep EPOS plan generation time as low as possible.
  4. **Sensor Field of View** – The field of view (degrees) in the cross-ground-track and along-ground-track directions on satellite sensor cameras. Set to (0.1, 0.1)

5. **Sensor Max Slew/Field of Regard –** The maximum amount (degrees) that sensor cameras can swivel in each direction. Set to (45, 45)
6. **Cloud Cover –** The amount of cloud cover over the battlefield. Set to None.

- **IBMS Fixed Parameters:**
  1. **Controller Type –** Which type of IBMS aircraft planner to use (either Heuristic or Optimal). Set to Heuristic.
  2. **Plan Horizon –** Length of time for which to plan when tasked. Set to 2 hours.
  3. **Number of non-ISR aircraft –** Set to 100 strike aircraft (60 of type 1, 40 of type 2), and 12 jammer aircraft
  4. **Number of type 1 ISR aircraft –** the number of type 1 ISR aircraft used in the scenario. Type 1 ISR aircraft are used primarily for locating targets although they have limited capability to ID and BDA targets. Set to 2.
  5. **Number and Location of Air Bases –** One base located ~350km east of battlefield.
  6. **Aircraft Average Speed**
     - Unmanned, ISR type 1 – 325kts
     - Unmanned, ISR type 2 – 150kts
     - Manned (non-ISR) – 450kts

- **ORSC Fixed Parameters:**
  1. **EVweaponeeringTable –** Table listing the expected number of EV hits necessary to destroy a target of a certain type. For each type, the number is as follows: WMD – 3, C3CP – 2, Fixed_target – 1, SAM – 2.
  2. **EVCrossRange –** The crossrange of EVs, once released from the LV. Set to 300km, which is larger than the distance between any two points on the battlefield. This essentially means there is no limitations on where an EV can strike.

3. **LV/EV speed –** Set so that it takes approximately 1.5 hours to travel from Vandenberg AFB to the target area.
4. **LVRecoveryTime –** The time it takes an LV to recover and be ready for another launch. Set to 12 hours from the time of the previous launch.
5. **Microsat Capable –** Whether ORSC has the capability of launching microsats into space. Currently, neither of the ORSC algorithms employ this capability.

- **SIM (primary) Fixed Parameters:**

1. **Campaign Length –** Length of battle campaign. Set to 86400 seconds (24 hours).
2. **Target Probability of Destruction –** (all weapons listed below are strike/fighter aircraft munitions except EV, which is the ORS munition):

Weapon: Type	LGB	JDAM	JSOW	HARM	AMSTE	EV
WMD	.5	.5	.75	.0	.9	.75
C3CP	.75	.5	.5	.0	.0	.50
Fixed_target	.5	.5	.5	.0	.0	.50
Fixed_SAM	.5	.5	.5	.5	.0	.75

3. **ISR Sensor Probability of correct ID –**

   - **ISR type 1 aircraft sensor**

Parameter: Type	Pcortype	Pcordmge
WMD	.30	.50
C3CP	.30	.50
Fixed_target	.50	.50
Fixed_SAM	.50	.50

   - **ISR type 2 aircraft sensor**

Parameter: Type	Pcortype	Pcordmge
WMD	.9	.9
C3CP	.9	.9
Fixed_target	.9	.9
Fixed_SAM	.9	.9

- **Satellite sensor**

```
   Parameter:    Pcortype  Pcordmge
Type
WMD                 .9       .9
C3CP                .9       .9
Fixed_target        .9       .9
Fixed_SAM           .9       .9
```

4. **SAM lethal radius –** Radius around a SAM in which it has the potential to destroy an aircraft. Set to 30km.
5. **Probability of Engagement with Jammers –** Given an aircraft is in a SAMs threat radius and has a jammer nearby, this is the probability that the SAM will fire a weapon at the aircraft. Set to 0.5
6. **Probability of Engagement without Jammers –** Same as above except without jammers. Set to 0.75.

5.2.2 Changeable Parameters

Throughout the experiments, certain parameters are modified from the values set in the Base Scenario (presented in Section 5.3.2) for testing purposes. This section outlines the parameters that are allowed to change and describes them briefly.

- **CC Changeable Parameters:**
  1. **ControllerType –** Whether the nominal or heuristic coordination algorithm is used.
  2. **RiskScaleParameter –** Proportion of ORSC target weight that is to come from estimated aircraft risk versus the commander's intent value. Value in [0,1].
  3. **IDImportance –** The commander's relative importance of target ID versus target BDA. Value in [0,1]
  4. **ISRDecayParameter –** Commander's estimate of how fast the values of ISR observations decay over time. Value in [0,1], where 0 means they do not decay at all and 1 means they decay very quickly.

- **EPOS Changeable Parameters:**
  1. **InScenario –** Whether satellites are used in the scenario.
  2. **NumSats –** The Number of satellites in the constellation. Value in {0, 36, 72}
  3. **MaxSlewRate –** Maximum degrees per second (in along-track and cross-track directions) the satellite camera sensors can slew. Value in {(1.0, 1.0), (3.0, 3.0), (5.0, 5.0)}.

- **IBMS Changeable Parameters:**
  1. **InScenario –** Whether aircraft are used in the scenario.
  2. **IPBMode –** This parameter quantifies how much target information is available at the start of the campaign. For a target that has no preliminary information given, it will have to be located, then ID'd, hit, and BDA'd. Otherwise, a target might already be located or located and ID'd at the start of the campaign. Values can be:
     - 0 – no preliminary target information (none located, none ID'd)
     - 1 – nominal (some located, some ID'd)
     - 2 – perfect information (All targets located and ID'd)}
  3. **NumType2ISRaircraft –** The number of type-2 ISR aircraft in the scenario. Type 2 ISR aircraft are used primarily for target ID and target BDA. Value in {10,...,15}

- **ORSC Changeable Parameters:**
  1. **InScenario –** Whether ORS assets are used in the scenario.
  2. **ControllerType –** Whether the heuristic or optimizing version of ORSC is used.
  3. **numSOV –** How many SOVs are used in the scenario. Value in {10,...,15}
  4. **RestrictToBarrier –** Whether SAMs in the barrier are given 10 times greater weight than any other target. Essentially, this causes ORSC to only

strike SAMs in the barrier *unless* there are no SAMs left or no SAMs that are ID'd.

- **SIM Changeable Parameters:**
  1. **TargetScenario** – Whether target scenario 1 or 2 is used. The target scenario defines the number, type, and locations of all targets on the battlefield. The locations of all targets are sampled from uniform distributions.
     - Target Scenario 1 has 8 WMD, 0 C3CP, 25 Fixed_targets, and 60 SAM (93 total). The WMD are located in the center 60% (lat/lon wise) of the battlespace. The Fixed_targets are spread throughout the battlefield. 35 of the SAMs are on the eastern border of the battlefield and serve as a barrier between the air base and the high-value targets. The remaining 25 SAMs are spread throughout the battlefield. Target Scenario 1 is shown in Figure 5-3.

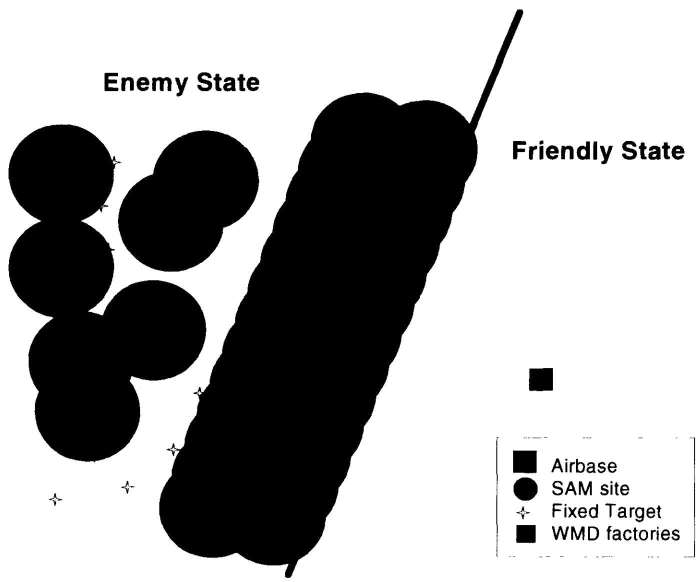

Figure 5-3: Target Scenario 1

     - Target Scenario 2 has 5 WMD, 10 C3CP, 35 Fixed_targets, and 50 SAM (100 total). The WMD are located in the center 75% (lat/lon wise) of the battlespace. The C3CP are in the center 90% of the battlefield. Both Fixed_targets and SAMs are spread throughout the battlefield. Target Scenario 2 is shown in Figure 5-4.

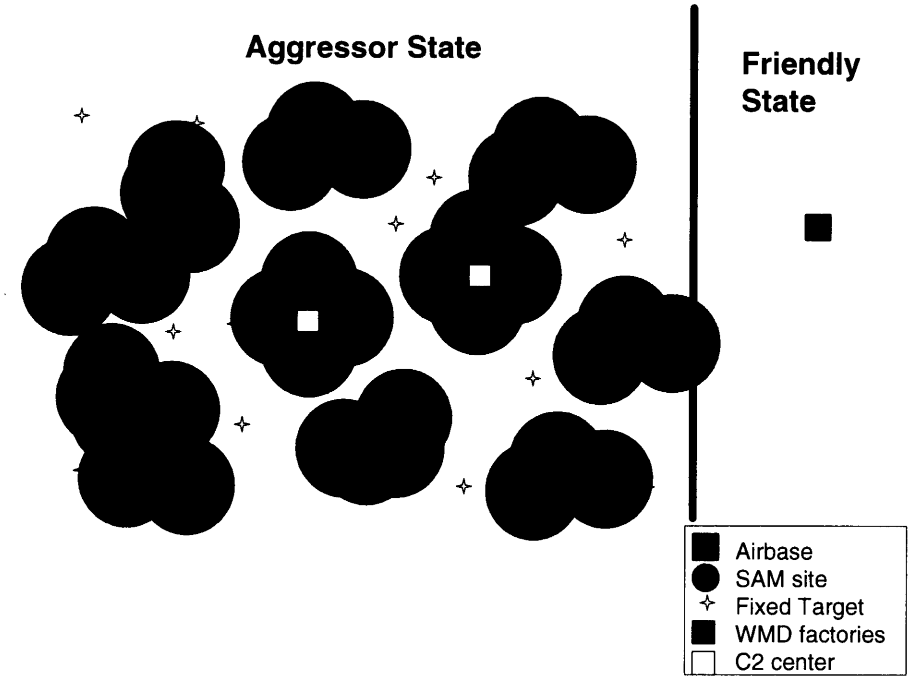

Figure 5-4: Target Scenario 2

2. **SAM_PDestruction** – Probability of a SAM destruction using any of the following: 1 Laser Guided Bomb (LGB), 2 Joint Direct Attack Munitions (JDAM), 1 Joint Standoff Weapon (JSOW), or 2 EVs. Value in [0,1]
3. **PattritGivenSAMEngagement** – Probability of aircraft attrition given engagement with a SAM. Value in [0,1].
4. **ReplanInterval** – The length of simulated time between controller plans (for all controllers). Value in {0.5 hours, 4 hours}.

5.2.3 Base Scenario Description

Here, we list the values of changeable parameters that define the base scenario. All parameters other than the ones listed are defined in the Fixed Scenario Attributes section. Additionally, we show results from one run of the base scenario for the purpose of familiarizing the reader with the type of output under examination.

BASE SCENARIO

- **CC Base Parameters:**
  1. **ControllerType =** Nominal
  2. **RiskScaleParameter** = 0.7
  3. **IDImportance** = 0.35
  4. **ISRDecayParameter** = 0.5

- **EPOS Base Parameters:**
  1. **InScenario** = No
  2. **NumSats** = 0
  3. **MaxSlewRate** = N/A

- **IBMS Base Parameters:**
  1. **InScenario** = Yes
  2. **IPBMode** = 1 (medium)
  3. **NumType2ISRAC** = 10

- **ORSC Base Parameters:**
  1. **InScenario** = No
  2. **ControllerType** = N/A
  3. **numSOV** = 0
  4. **RestrictToBarrier**= N/A

- **SIM Changeable Parameters:**
    1. **TargetScenario** = Scenario 1
    2. **SAM_PDestruction** = 0.75
    3. **PattritGivenSAMEngagement** = 0.01
    4. **ReplanInterval** = 0.5 hours

The base scenario is used to analyze the effectiveness of aircraft alone in the context of target scenario 1. An overview of how the base scenario performed is presented in Table 5-1.

Basic Performance Statistics		
Number of Targets Destroyed Above Threshold	**50**	tgts out of 93
Targets First Hit by Type:		
Aircraft	50	tgts
EV	0	tgts
Aggregate Damage Fraction (ADF)	**0.466**	
Time to Achieve 0.85 ADF	Never	hours
Fraction High-Val Targets Destroyed	0.5	
Num Targets BDA'd	**18**	tgts
BDA'd by type:		
ISRtype2	18	tgts
Satellite	0	tgts
Avg Time from ID to Hit	4.98	hours
Avg Time from Hit to BDA	**10.14**	hours
Num AC Sorties	**380**	sorties
Num EV Sorties	0	sorties
Percent ADF per Sortie (incl. EV sorties)	0.123%	dmg/sortie
Percent Attrition per Sortie	3.947%	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**1.316%**	attrit/sortie
Num Aircraft Attritions	15	attrit
Attritions by type:		
Manned	5	attrit
Unmanned	10	attrit

Table 5-1: AC Statistics

Notice that only **50 targets out of 93 are destroyed** above the damage threshold. The **Aggregate Damage Fraction (ADF), which is a weighted measure of total target destruction, is only 0.466**. The ADF is weighted according to initial commander's intent values and so a WMD destruction adds to this quantity more than a SAM or Fixed_target destruction. Notice that **50% of the high-value targets (WMD) are destroyed**. Only **18 out of 50 of the destroyed targets are BDA'd**. The average time from the first hit on a target hit to the first BDA is over 10 hours (if a target is hit but never BDA'd this quantity is equal to the time from first hit till the end of the campaign). There are 380 total sorties flown in this campaign and the aircraft suffer a **1.32% manned attrition rate**, (percent attritions per sortie) where the ISR aircraft are assumed unmanned. In Figure 5-5, the ADF over the course of the campaign is shown, broken down by target category:

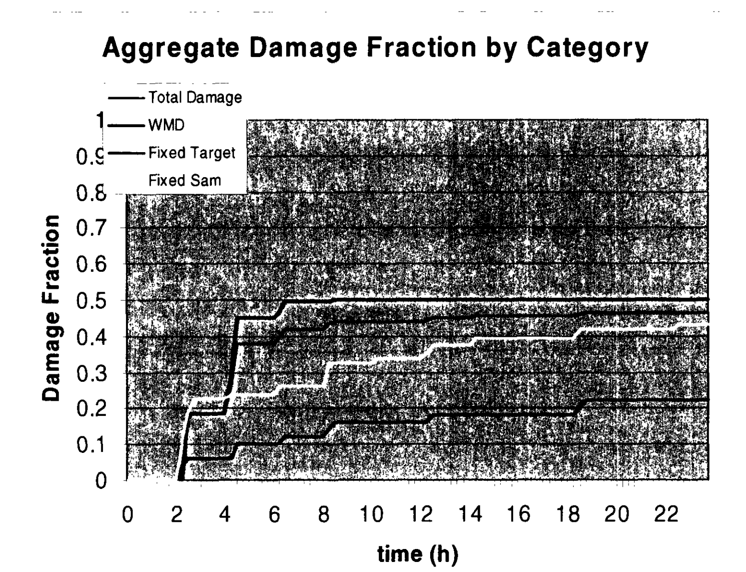

Figure 5-5: Target Damage by Category

The diminishing marginal returns on target damage is common in battlefield operations although one question that arises from this figure is why there is no additional WMD damage after the 8th hour. This is because the type 2 ISR aircraft get shot down at a fairly fast rate (all but two by the 10th hour). IBMS can only attack targets that have been ID'd and, in fact, half of the WMD targets never got ID'd.

The aircraft attritions by type are shown in Figure 5-6. The reason ISR type 2 aircraft have such a high attrition rate is simply because they remain over the battlefield for the entire time and do not have jammer escorts. Fighter and jammer aircraft ingress and egress the battlespace only when they are attacking a target. ISR type 1 aircraft (not shown) will never have attritions because they remain well clear of the battlespace and are able to perform their wide-area searches at much greater distances.

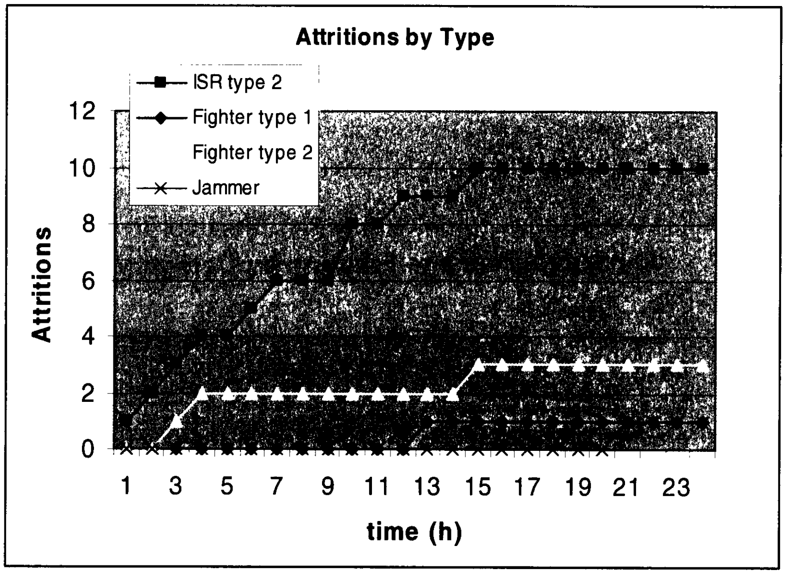

Figure 5-6: Aircraft attritions by Type

Lastly, we describe the Kill Chain Progression for the base scenario (shown in Figure 5-7). The Kill Chain Progression lists the order that target states follow in battlespace operations and, for our purposes, are (ID, Hit, BDA). As discussed previously, a target must be ID'd before being hit and hit before being BDA'd. Figure 5-7 shows the total number of targets that have been ID'd at least once, damaged above the threshold, and BDA'd at least once (respectively). The number of targets ID'd reaches 82, which is only 11 short of the total on the battlefield. So even though the type 2 ISR aircraft are attrited fairly early, the type 1 ISR aircraft are able to compensate fairly well. The number of targets damaged above the threshold reaches 50, or about 54% of the total. The number of targets BDA'd at the end of the campaign is only 18. We can infer that once all of the ISR type 1 aircraft are attrited, there is no hope for attaining additional BDA's. Thus, the ISR type 1 attrition rate severely limits performance in the base scenario in this way. In Figure 5-7, which shows the Kill Chain Progression, each category's value is read from the x-axis up to the line. So the number of targets ID'd at the end of the campaign is 82, not 82 minus the number of targets destroyed.

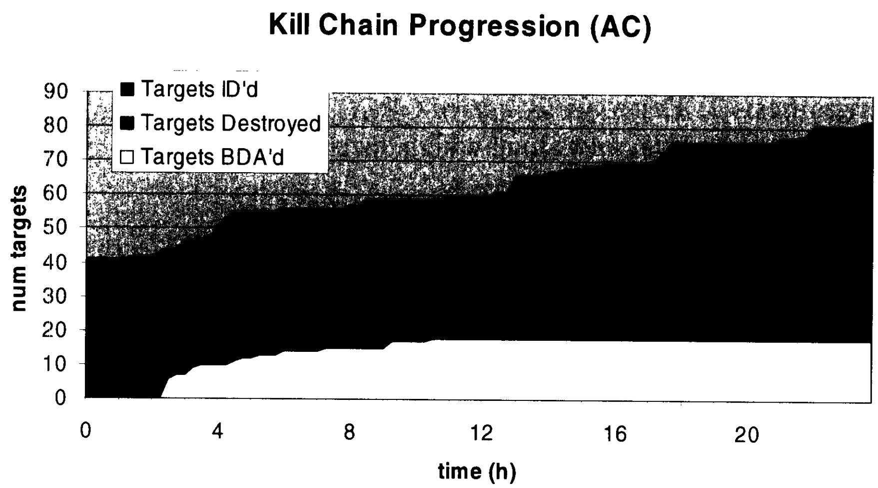

Figure 5-7: Kill Chain Progression

5.3 Primary Experiments

The methodology we use to test the *CDASOCS* system is to formulate a series of hypotheses about system behavior, perform run sequences to test each hypothesis, and analyze the simulation results. Each run uses a different set of parameters that define the scenario context and algorithm type. Normally for Monte Carlo simulation, it is recommended that the number of runs used to estimate a distribution mean be absolutely no lower than five and generally at least 30. However, due to technical problems beyond the scope of this thesis, the implemented *CDASOCS* currently only runs to completion about 20% of the time. This has significantly limited the potential for a large number of runs, but after repeated testing, it is evident that the simulation is fairly insensitive to perturbations. Thus, we proceed by performing one run of each of the scenarios listed below. For each experiment, we state the experiment title, describe the hypothesis, list the scenarios used for comparison, and discuss the conclusions.

5.3.1 Experiment 1: Addition of ORSC to Base Scenario

- **Hypothesis 1:** The *addition* of ORS assets over the campaign causes an increase in total targets destroyed and aggregate damage fraction.

- **Hypothesis 2:** The *coordination* of ORSC with AC causes an additional increase in these metrics along with a decrease in manned aircraft attrition rate.

- **Scenarios Involved:**
  A. Base Scenario

  B. Addition of ORSC with:
     - CCControllerType = Nominal
     - ORSControllerType = Heuristic
     - numSOV = 10
     - RestrictToBarrier = No

  C. Addition of ORSC with:
     - CCControllerType = Heuristic
     - ORSControllerType = Heuristic
     - numSOV = 10
     - RestrictToBarrier = Yes

- **Results:** In each figure below, the results of all three scenarios are shown. First, Figure 5-8 shows the time progression of the Aggregate Damage Fraction (ADF) metric. At time 24 (the end of the campaign), Scenario A has an ADF of 0.466, Scenario B has an ADF of 0.572, and Scenario C has an ADF of 0.653. Just as hypothesized, the addition of ORSC and the coordination of AC with ORSC each yield successively higher damage fractions. The addition of ORSC yields a 10.6% improvement while the coordination of AC and ORSC yields an additional 8.1% improvement. It should be pointed out that Scenario B (AC and ORSC) does start out with a higher rate of damage than Scenario C. The reason is because, in the uncoordinated version, ORS assets are used to attack the highest value targets. This is in contrast to the coordinated version where ORSC has the competing objectives of attacking some high-value targets and some targets that pose a threat to aircraft. In the end though, the coordinated version wins out in terms of total damage.

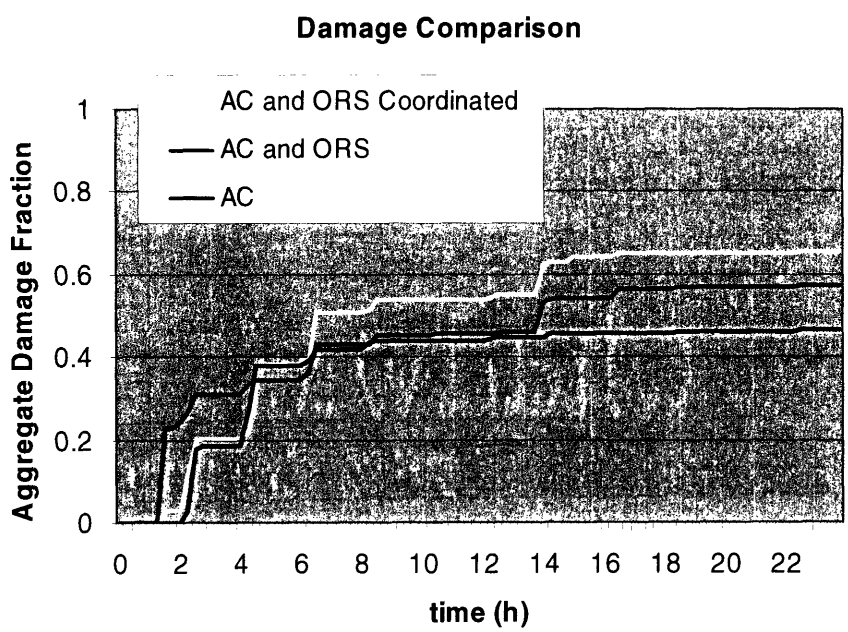

Figure 5-8: Damage Comparison for Experiment 1

Next, Table 5-2 shows the basic performance statistics of each scenario. As hypothesized, each successive scenario brings a higher number of targets destroyed, as well as fraction of high-value targets destroyed. In addition, the manned aircraft attrition rate decreases slightly with each scenario. This result is in line with hypothesis #2 and illustrates the fact that when AC and ORSC resources are coordinated (vs. uncoordinated), it is possible to achieve greater damage while suffering a lower manned attrition percentage. The attrition percentage is not shown over time, as it is intended to be a measure of overall campaign attrition loss rate. Finally, it is noted that there are six more targets BDA'd in scenario C than in the other two scenarios. This could be from better information flow in the coordinated scenario, or it could just be because more targets were struck and therefore more had a chance of being BDA'd.

Basic Performance Statistics				
	AC Alone	AC and ORS	AC and ORS Coord.	
Number of Targets Destroyed Above Threshold	50	53	63	tgts out of 93
Targets First Hit by Type:				
Aircraft	50	34	42	tgts
EV	0	19	21	tgts
Aggregate Damage Fraction (ADF)	0.466	0.572	0.653	
Time to Achieve 0.85 ADF	Never	Never	Never	hours
Fraction High-Val Targets Destroyed	0.5	0.625	0.75	
Num Targets BDA'd	18	18	24	tgts
BDA'd by type:				
ISRtype2	18	18	24	tgts
Satellite	0	0	0	tgts
Avg Time from ID to Hit	4.98	4.08	5.32	hours
Avg Time from Hit to BDA	10.14	11.79	9.83	hours
Num AC Sorties	380	389	407	sorties
Num EV Sorties	0	72	51	sorties
Percent ADF per Sortie (incl. EV sorties)	0.123%	0.147%	0.160%	dmg/sortie
Percent Attrition per Sortie	3.947%	3.856%	3.686%	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	1.316%	1.028%	0.983%	attrit/sortie
Num Aircraft Attritions	15	13	13	
Attritions by type:				
Manned	5	4	4	attrit
Unmanned	10	9	9	attrit

Table 5-2: Performance Stats Comparison for Experiment 1

5.3.2 Experiment 2: Addition of SATC to Base Scenario

- **Hypothesis 1:** The *addition* of satellites over the campaign causes a decrease in the average time from the first hit of a target to the first BDA and an increase in the rate of target ID's and number of targets destroyed.

- **Hypothesis 2:** The *coordination* of satellites over the campaign causes an additional decrease in the average time from first hit to first BDA.

- **Scenarios Involved:**

  A. Base Scenario

  B. Addition of SATC with:
  - CCControllerType = Nominal
  - NumSats = 36
  - MaxSlewRate = (3.0, 3.0) deg/sec

  C. Addition of SATC with:
  - CCControllerType = Heuristic
  - NumSats = 36
  - MaxSlewRate = (3.0, 3.0) deg/sec

o **Results:** Table 5-3 shows the basic performance statistics for Experiment #2. Just as expected, the addition of satellites brings a dramatic decrease in the average BDA time and a dramatic increase in the number of targets destroyed. The increase in targets destroyed comes most likely as a result from a faster rate of target ID's (as shown in Figures 5-9 and 5-10). As targets become ID'd quicker, AC is able to assign more attack missions against these targets, leading to an increase in targets destroyed. Also, as hypothesized, the coordination of AC and SATC brings about an additional reduction in average BDA time (0.67 hours quicker).

Basic Performance Statistics				
	AC Alone	**AC and SAT**	**AC and SAT Coord.**	
Number of Targets Destroyed Above Threshold	**50**	**84**	**82**	tgts out of 93
Targets First Hit by Type:				
Aircraft	**50**	**85**	**82**	tgts
EV	**0**	**0**	**0**	tgts
Aggregate Damage Fraction (ADF)	**0.466**	**0.899**	**0.801**	
Time to Achieve 0.85 ADF	**Never**	**10.5**	**Never**	hours
Fraction High-Val Targets Destroyed	**0.5**	**1**	**0.875**	
Num Targets BDA'd	**18**	**77**	**68**	tgts
BDA'd by type:				
ISRtype2	**18**	**34**	**26**	tgts
Satellite	**0**	**43**	**42**	tgts
Avg Time from ID to Hit	**4.98**	**3.21**	**2.85**	hours
Avg Time from Hit to BDA	**10.14**	**5.33**	**4.66**	hours
Num AC Sorties	**380**	**679**	**624**	sorties
Num EV Sorties	**0**	**0**	**0**	sorties
Percent ADF per Sortie (incl. EV sorties)	**0.123%**	**0.132%**	**0.128%**	dmg/sortie
Percent Attrition per Sortie	**3.947%**	**2.209%**	**2.404%**	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**1.316%**	**1.325%**	**0.962%**	attrit/sortie
Num Aircraft Attritions	**15**	**19**	**15**	
Attritions by type:				
Manned	**5**	**9**	**6**	attrit
Unmanned	**10**	**10**	**9**	attrit

Table 5-3: Performance Stats Comparison for Experiment 2

It is interesting to note that the coordinated scenario fails to beat the uncoordinated scenario in terms of target destructions, the ADF, and targets BDA'd. This is likely the result of too high of a weight being given toward SATC BDA. In other words, too much emphasis is placed on the function of satellites performing BDA and not enough on satellites performing ID on new targets. While this philosophy does result in a quicker *average* BDA time, it hampers the *rate* of target ID's, rate of target destructions, and as a result, total number of BDA's.

In Figures 5-9, 5-10, and 5-11, we show the kill-chain progression of each scenario in Experiment #2. The ideal figure is for all three curves to have as big a slope and to be as close together as possible because this would indicate that a) targets are destroyed and BDA'd quickly and b) targets go from one state to another in close succession.

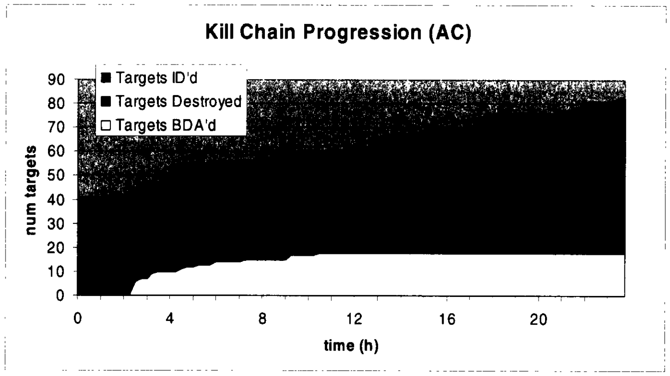

Figure 5-9: Kill Chain Progression for AC only Scenario

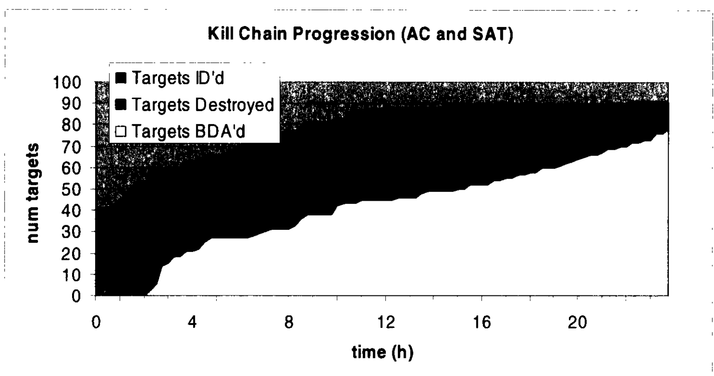

Figure 5-10: Kill Chain Progression for AC and SATC Scenario

Notice the increased slope in the target ID line between figure 5-9 and 5-10. Also note in figure 5-11 how the target BDA curve stays closer to the targets destroyed curve than in figure 5-10 (this is more prevalent in the first 12 hours). This is the result of satellites and predators working more collaboratively in the coordinated scenario.

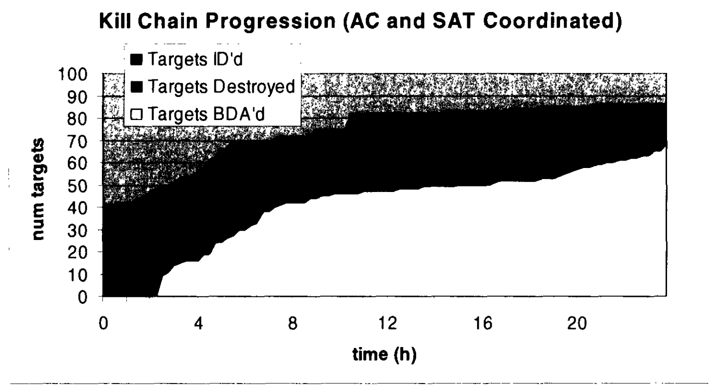

Figure 5-11: Kill Chain Progression for AC and SATC Coordinated Scenario

5.3.3 Experiment 3: Addition of Both ORSC and SATC to Base Scenario

- **Hypothesis 1:** The *addition* of ORS assets and satellites over the campaign causes: 1) an increase in targets destroyed and ADF, and 2) a decrease in the average time from the first hit of a target to the first BDA.

- **Hypothesis 2:** The *coordination* of ORSC, SATC, and AC over the campaign causes: 1) a further increase in targets destroyed and ADF, 2) a decrease in the manned aircraft attrition rate, 3) a further decrease in the average time from first hit to first BDA, 4) a kill-chain progression in closer succession, meaning that at any given time, the number of targets hit and BDA'd is close to the number of targets ID'd.

- **Scenarios Involved:**

  A. Base Scenario

  B. Addition of ORSC and SATC with:
  - CCControllerType = Nominal
  - ORSControllerType = Heuristic
  - numSOV = 10
  - RestrictToBarrier = No
  - NumSats = 36

- MaxSlewRate = (3.0, 3.0) deg/sec

C. Addition of ORSC and SATC with:
- CCControllerType = Heuristic
- ORSControllerType = Heuristic
- numSOV = 10
- RestrictToBarrier = Yes
- NumSats = 36
- MaxSlewRate = (3.0, 3.0) deg/sec

- **Results:** Table 5-4 shows the Basic Performance Statistics for Experiment #3. *Adding* the two space controllers caused an increase in the number of targets destroyed, fraction of high-value targets destroyed, ADF, and the number of BDAs and caused a decrease in average BDA time and manned aircraft attrition rate. Furthermore, *coordinating* among resources provided a 7.1% increase in number of targets destroyed, a 3.7% increase in ADF, an 8.2% increase in number of targets BDAd, a 5.8% decrease in average BDA time, and a 44% decrease in the manned aircraft attrition rate.

Basic Performance Statistics				
	AC Alone	**AC, ORS, SAT**	**AC, ORS, SAT Coord.**	
Number of Targets Destroyed Above Threshold	50	84	90	tgts out of 93
Targets First Hit by Type:				
Aircraft	50	66	59	tgts
EV	0	18	32	tgts
Aggregate Damage Fraction (ADF)	0.466	0.884	0.921	
Time to Achieve 0.85 ADF	Never	10.75	20.5	hours
Fraction High-Val Targets Destroyed	0.5	1	1	
Num Targets BDA'd	18	73	79	tgts
BDA'd by type:				
ISRtype2	18	38	18	tgts
Satellite	0	35	61	tgts
Avg Time from ID to Hit	4.98	2.78	2.92	hours
Avg Time from Hit to BDA	10.14	5.01	4.72	hours
Num AC Sorties	380	642	619	sorties
Num EV Sorties	0	72	74	sorties
Percent ADF per Sortie (incl. EV sorties)	0.123%	0.138%	0.149%	dmg/sortie
Percent Attrition per Sortie	3.947%	2.336%	2.423%	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	1.316%	1.090%	0.646%	attrit/sortie
Num Aircraft Attritions	15	16	14	
Attritions by type:				
Manned	5	7	4	attrit
Unmanned	10	9	10	attrit

Table 5-4: Basic Performance Statistics for Experiment 3

The only adverse feature of the coordinated scenario is the amount of time it took to achieve 85% target damage (down 9.75 hours from the uncoordinated scenario). This stems from the fact that the uncoordinated scenario is more of a greedy algorithm whereas the coordinated version has the additional objective of protecting aircraft.

Next, the kill-chain progression for each of the last two scenarios is shown (the kill-chain progression for scenario A is the same as that in Figure 5-9). In the case with all sub-controllers uncoordinated, targets are actually ID'd at a faster rate than in the coordinated case. The reason is that in the coordinated version, SATC is concerned with target BDA in addition to target ID.

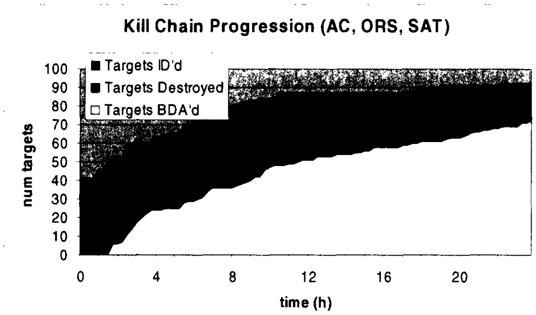

Figure 5-12: Kill Chain Progression for AC, ORSC, and SATC Scenario

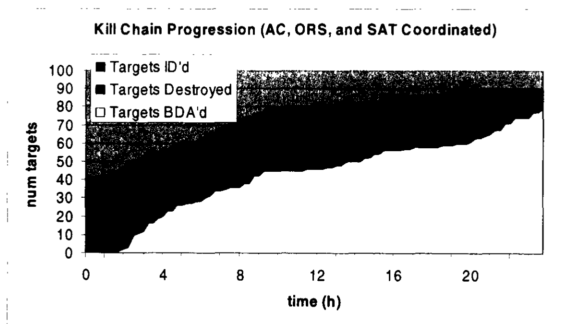

Figure 5-13: Kill Chain Progression for AC, ORSC, and SATC Coordinated Scenario

Secondly, consider the two BDA lines in Figures 5-12 and 5-13 from time 12 onward. Notice that the rate of BDA in the uncoordinated case is fairly linear while that in the coordinated case takes more of an exponential shape. One explanation for the exponentially shaped BDA curve is that, at time 12, most of the ISR type 2 aircraft are already attrited leaving SATC to do all target BDA and ID. As more and more targets become ID'd, SATC can focus more and more on BDA, hence, the exponential shaped curve.

In Figures 5-14 and 5-15, the number of targets remaining and ADF are shown. In both figures, the uncoordinated case starts out beating the coordinated case but finishes behind at the end of the campaign. Part of this is, once again, due to the fact that the uncoordinated case follows more of a greedy methodology, where each controller attempts to gain the highest value the quickest, negating any possibilities of collaboration. It should be noted once again that the coordinated case finishes the campaign with more targets destroyed, a higher ADF, more targets BDA'd, faster average BDA time, and less total as well as manned aircraft attritions.

Another reason why the rate of target destruction is initially higher in the uncoordinated case has to do with the dependency of target destruction on target ID. Thus, the coordinated case, which has slightly lower target ID rate (due to increased BDA importance in SATC), is bound to lag behind in target destruction rate until the number of IDs can catch up.

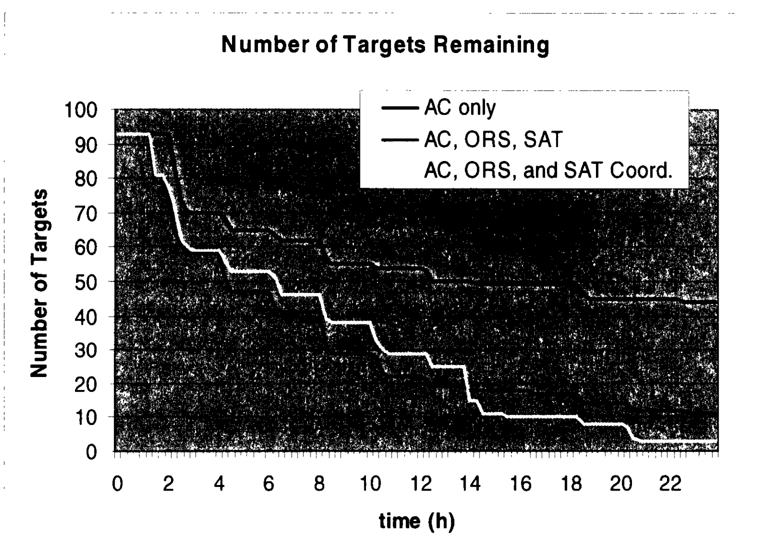

Figure 5-14: Number of Targets Remaining for Experiment 3

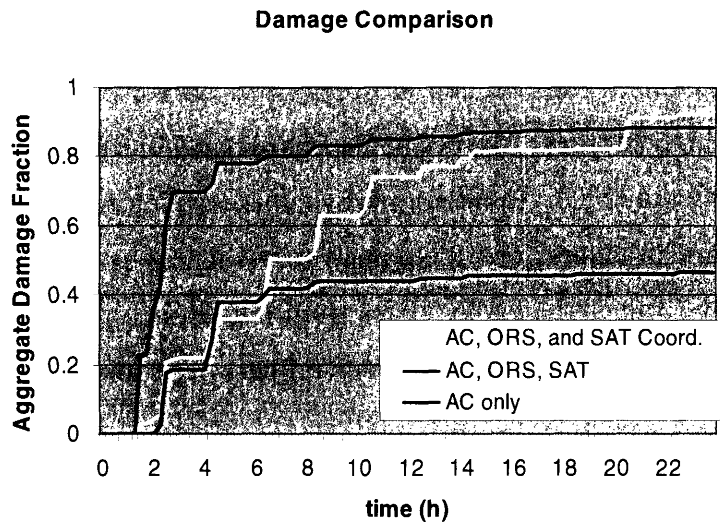

Figure 5-15: Damage Comparison for Experiment 3

Finally, we compare the resource allocation of the two scenarios that involve all three sub-controllers. The main difference, as seen in Figure 5-16, is that EVs are used to attack most of the WMDs in the uncoordinated scenario (hence why they gain such a large value at first). This is in contrast to the coordinated scenario where EVs are primarily used against SAMs to clear the way for aircraft to attack the higher value targets.

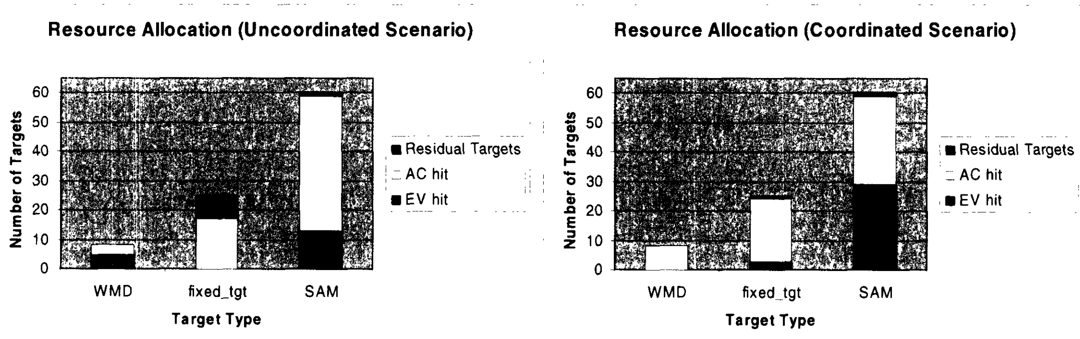

Figure 5-16: Comparison of Resource Allocation for Experiment 3

5.4 Parametric Variations

In this section, we explore parameter variations in all five of the functional components of *CDASOCS.* All tests in this section are performed on the coordinated scenario and involve all three sub-controllers. We define the coordinated scenario as the Scenario C of Experiment 3 in section 5.3.3. There are no hypotheses given for these experiments, as they are exploratory in nature. The motivation for using the specific parameters tested here, along with the values they take, arises from interesting questions and conjectures that were thought up while conducting the three primary experiments.

5.4.1 CC/ORSC Parameter Variations

- **Experiment 4: Variations on which targets to assign to ORSC**
  - **Discussion:** In Experiment 3C, CC uses ORSC exclusively to attack targets in the barrier region (comprising the eastern border of the battlefield). Here, we relax this constraint and also vary the RiskScaleParameter. The RiskScaleParameter defines how much of the ORSC weight comes from the estimated target risk to aircraft (versus from commander's intent). Thus, lower values of this parameter should result in ORSC attacking higher-value targets whereas higher values should result in ORSC attacking higher-risk parameters.
  - **Scenarios Involved:**

    A. Coordinated Scenario with:
       - RiskScaleParameter = 0.4
       - RestrictToBarrier = No

    B. Same as Scenario A with RiskScaleParameter = 0.6
    C. Same as Scenario A with RiskScaleParameter = 0.8

  - **Results:** In Scenario A, ORSC attacks 3 of the 8 WMDs. In Scenario B, ORSC attacks 2 WMDs, and in Scenario C, none. Just as we would expect, Scenario A, B, and C have successively lower *initial* rates of target damage (ADF), although at the end of the campaign, the three scenarios have virtually the same ADF. This is shown in Figure 5-17.

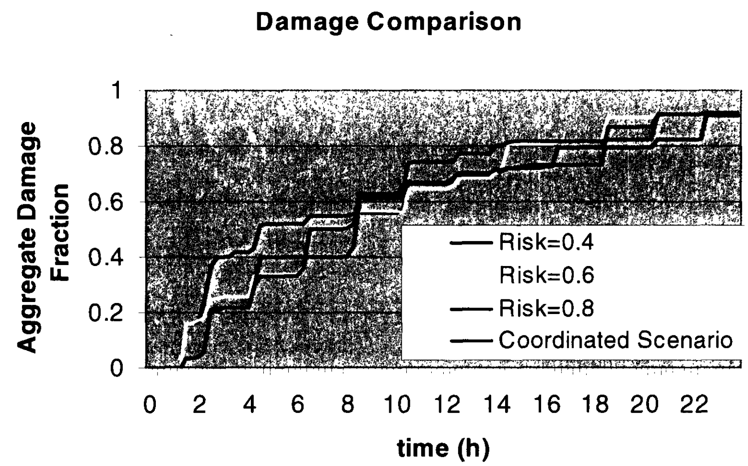

Figure 5-17: ADF Comparison for Experiment 4

It should also be pointed out that the coordinated scenario has a lower rate of manned aircraft attrition than in each of these scenarios. There is, on average, 0.37% less attrition per sortie in the coordinated scenario. One reason is that in the coordinated scenario, where ORSC assets are restricted to only attack targets in the barrier, the aircraft gain safer routes and thus have a lower probability of coming in contact with a SAM (unless they are purposefully attacking it).

- **Experiment 5: Increasing the number of LVs in ORSC**

  - **Discussion:** The purpose of this test is to investigate the effects of increasing the number of LVs.

  - **Scenarios Involved:**

    A. Coordinated Scenario with numSOVs = 10
    B. Coordinated Scenario with numSOVs = 15

- **Results:** The Basic Performance Statistics comparison is shown in Table 5-5 for the two scenarios. As can be seen in the figure, Scenario B does better in almost every category. The main points of interest are the decrease of 6 hours for the time to achieve 0.85 ADF and the decrease of the Percent Manned Aircraft Attrition per Sortie (almost cut in half).

Basic Performance Statistics			
	numLVs = 10	**numLVs = 15**	
Number of Targets Destroyed Above Threshold	**90**	**92**	tgts out of 93
Targets First Hit by Type:			
Aircraft	**59**	**49**	tgts
EV	**32**	**36**	tgts
Aggregate Damage Fraction (ADF)	**0.921**	**0.926**	
Time to Achieve 0.85 ADF	**20.5**	**14.5**	hours
Fraction High-Val Targets Destroyed	**1**	**1**	
Num Targets BDA'd	**79**	**88**	tgts
BDA'd by type:			
ISRtype2	**18**	**34**	tgts
Satellite	**61**	**54**	tgts
Avg Time from ID to Hit	**2.92**	**2.01**	hours
Avg Time from Hit to BDA	**4.72**	**3.74**	hours
Num AC Sorties	**619**	**565**	sorties
Num EV Sorties	**74**	**110**	sorties
Percent ADF per Sortie (incl. EV sorties)	**0.149%**	**0.164%**	dmg/sortie
Percent Attrition per Sortie	**2.423%**	**2.655%**	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**0.646%**	**0.354%**	attrit/sortie
Num Aircraft Attritions	**14**	**11**	
Attritions by type:			
Manned	**4**	**2**	attrit
Unmanned	**10**	**9**	attrit

Table 5-5: Basic Performance Statistics for Experiment 5

- **Experiment 6: Increasing the Importance of Target ID (vs. BDA)**

  - **Discussion:** In the coordinated scenario, the rate of target ID lags behind that of the uncoordinated scenario and this is the potential cause of a lag in target BDA further on in the campaign. Hence, it is our goal here to determine the effect of increasing the importance of the function of target ID (versus target BDA) in the CC algorithm. Doing this should cause CC to assign more targets that need ID to EPOS and less targets that need BDA.

  - **Scenarios Involved:**

A. Coordinated Scenario with IDImportance = 0.35
B. Coordinated Scenario with IDImportance = 0.45

Results: The Basic Performance Statistics comparison for this test is shown in Table 5-6. Notice that Scenario B reduces the time to achieve 0.85 ADF by 12 hours. Also note that Scenario B, which is move away from BDA importance, actually causes *more* target BDA's in the end (84 vs. 79). An explanation is that targets are destroyed at a faster rate and thus have more time to be BDA'd before the end of the campaign. There is a tradeoff between time to do BDA and attrition though. There is a much higher average BDA time (an increase of 1.88 hours) and a much higher rate of manned aircraft attrition (9 more manned aircraft are lost in the second scenario). The increased attritions are likely due to the fact that since targets are BDA'd at a much slower rate, IBMS is not as likely to know which SAMs were destroyed by LVs (IBMS only gains this information after target BDA). Without knowledge of SAM destructions, IBMS would not try to route its planes through the safe corridors created by LVs and thus would experience greater attrition.

Basic Performance Statistics			
	IDImportance = 0.35	IDImportance = 0.45	
Number of Targets Destroyed Above Threshold	90	90	tgts out of 93
Targets First Hit by Type:			
Aircraft	59	64	tgts
EV	32	27	tgts
Aggregate Damage Fraction (ADF)	0.921	0.919	
Time to Achieve 0.85 ADF	20.5	8.5	hours
Fraction High-Val Targets Destroyed	1	1	
Num Targets BDA'd	79	84	tgts
BDA'd by type:			
ISRtype2	18	34	tgts
Satellite	61	50	tgts
Avg Time from ID to Hit	2.92	2.57	hours
Avg Time from Hit to BDA	4.72	6.50	hours
Num AC Sorties	619	628	sorties
Num EV Sorties	74	71	sorties
Percent ADF per Sortie (incl. EV sorties)	0.149%	0.146%	dmg/sortie
Percent Attrition per Sortie	2.423%	2.389%	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	0.646%	2.070%	attrit/sortie
Num Aircraft Attritions	14	20	
Attritions by type:			
Manned	4	13	attrit
Unmanned	10	7	attrit

Table 5-6: Basic Performance Statistics for Experiment 6

Ideally, a coordinated algorithm should pass information to IBMS indicating which targets ORSC is *planning* to attack, which would subvert this problem. However, software constraints currently prevent this type of information sharing.

In Figures 5-18 and 5-19 we show the kill-chain progressions of the two scenarios. This illustrates that a higher IDImportance means targets get ID'd at a faster rate but at the expense of a BDA rate significantly lagging behind target destructions.

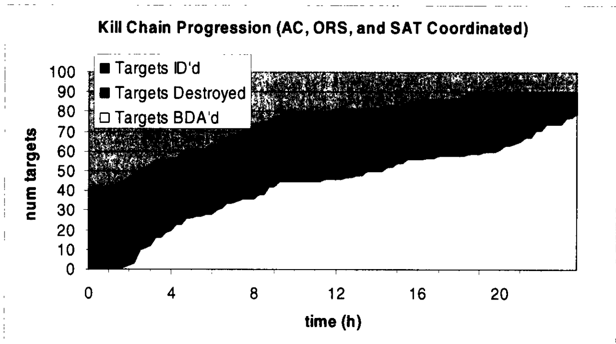

Figure 5-18: Kill Chain Progression for Scenario A in Experiment 6

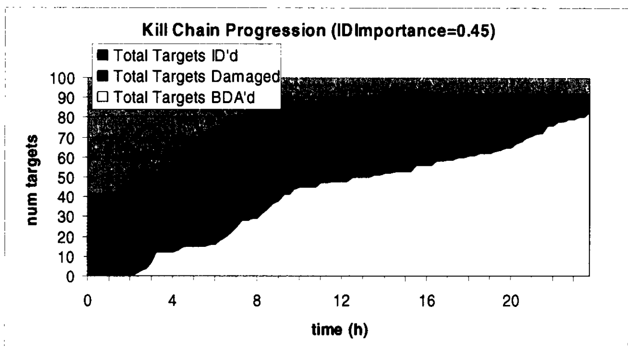

Figure 5-19: Kill Chain Progression for Scenario B in Experiment 6

Lastly, in Figure 5-20, we compare the ADFs of the two scenarios over the course of the campaign. Although the Coordinated Scenario does finish slightly above the scenario with higher IDImportance, it is behind for the majority of the campaign. With a higher IDImportance, targets get ID'd at a faster rate and therefore may be destroyed at a faster rate.

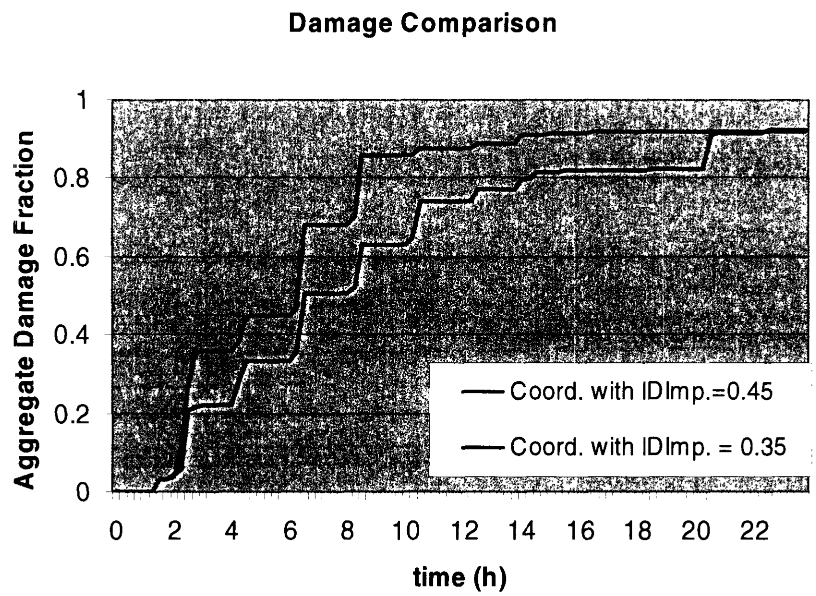

Figure 5-20: ADF Comparison for Experiment 6

5.4.2 EPOS Parameter Variations

- **Experiment 7: Increasing the number of satellites**
  - **Discussion:** In the first of the EPOS tests, we increase the number of satellites from 36 to 72 to explore the effects of doubling satellite assets.
  - **Scenarios Involved:**

    A. Coordinated Scenario with numSats = 36
    B. Coordinated Scenario with numSats = 72

  - **Results:** In Table 5-7, the Basic Performance Statistics of the two scenarios are shown. Just as in the case with an increased number of LVs, the additional satellites produce improvement in almost every area. Especially relevant are the decrease in time to achieve 0.85 ADF, the increased number of targets BDA'd and the decrease in the average BDA time. Also, we point out that the manned aircraft attrition rate rose fairly significantly. This is attributed to the fact that aircraft were used to hit relatively more targets in the case with 72 satellites – which in turn may be attributed to the fact that information is updated quicker in the 72 satellite case and so AC uses that information to act before ORSC can utilize all of its resources.

Basic Performance Statistics			
	numSATS = 36	**numSATS = 72**	
Number of Targets Destroyed Above Threshold	**90**	**92**	tgts out of 93
Targets First Hit by Type:			
Aircraft	**59**	**68**	tgts
EV	**32**	**25**	tgts
Aggregate Damage Fraction (ADF)	**0.921**	**0.923**	
Time to Achieve 0.85 ADF	**20.5**	**14.5**	hours
Fraction High-Val Targets Destroyed	**1**	**1**	
Num Targets BDA'd	**79**	**92**	tgts
BDA'd by type:			
ISRtype2	**18**	**19**	tgts
Satellite	**61**	**73**	tgts
Avg Time from ID to Hit	**2.92**	**2.88**	hours
Avg Time from Hit to BDA	**4.72**	**2.79**	hours
Num AC Sorties	**619**	**633**	sorties
Num EV Sorties	**74**	**76**	sorties
Percent ADF per Sortie (incl. EV sorties)	**0.149%**	**0.146%**	dmg/sortie
Percent Attrition per Sortie	**2.423%**	**2.370%**	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**0.646%**	**1.106%**	attrit/sortie
Num Aircraft Attritions	**14**	**17**	
Attritions by type:			
Manned	**4**	**7**	attrit
Unmanned	**10**	**10**	attrit

Table 5-7: Basic Performance Statistics Comparison for Experiment 7

- **Experiment 8: Varying the satellite sensor slew rate**

  - **Discussion:** Here, we examine the effects of varying the rate at which satellite sensors can slew, which is nominally set to 3.0 degrees/second in each direction.

  - **Scenarios Involved:**

A. Coordinated Scenario with MaxSlewRate = (3.0, 3.0)
B. Coordinated Scenario with MaxSlewRate = (1.0, 1.0)
C. Coordinated Scenario with MaxSlewRate = (5.0, 5.0)

- **Results:** Although the data produced by these scenarios was slightly different, there are no significant trends in this test. An increased sensor slew rate should increase the capability of each satellite but the perturbations caused by these scenarios are most probably too small to have much effect. It could be that each satellite is over the battlefield for such a short period of time that an increased slew rate does not necessarily cause an increase in the number of potential target looks.

5.4.3 IBMS Parameter Variations

- **Experiment 9: Increasing the number of ISR type 2 aircraft**

  - **Discussion:** The first IBMS parameter variation is an increase in the number of ISR type 2 aircraft. In the coordinated scenario, it becomes apparent that all of these aircraft, which are designed to do target ID and BDA, are attrited fairly quickly. One interesting question is to determine the effect of increasing the number of these aircraft from 10 to 15.

  - **Scenarios Involved:**

    A. Coordinated Scenario with numISRType2AC = 10
    B. Coordinated Scenario with numISRType2AC = 15

  - **Results:** Increasing the number of ISR type 2 aircraft actually causes the ADF to decrease from 0.921 in Scenario A to 0.829 in Scenario B. At first, this seems quite dramatic. It should be noted, however, that only one less target is destroyed in Scenario B than in Scenario A. It happens to be that the one less target is a high-value WMD and has a significant effect on ADF. At any rate, performance in Scenario B is not better than in Scenario A. One explanation is that the presence of more ISR type 2 aircraft initially causes IBMS to act as if it is more capable than it is. For example, by the fifth hour of the campaign, Scenario A still has 7 ISR type 2 aircraft remaining. In contrast, Scenario B only has 5 remaining ISR type 2 aircraft by the fifth hour. Scenario B certainly cannot be expected to perform better from the fifth hour onward.

- **Experiment 10: Varying the level of initial target knowledge**

  - **Discussion:** The reason this parameter is placed here and not the SIM tests is simply because the sub-controllers all act off of the information model contained in IBMS. There are three levels of initial target knowledge, or IPBMode (0 = none, 1 = nominal, 2 = perfect). All scenarios up to this point use IPBMode = 1. Here, we investigate what happens when the sub-controllers begin with no initial target knowledge (IPBMode = 0)

  - **Scenarios Involved:**

    A. Coordinated Scenario with IPBMode = 1
    B. Coordinated Scenario with IPBMode = 0

  - **Results:** As would be expected, starting the campaign with no initial target knowledge inhibits the overall success of the coordinated scenario even though approximately the same number of aircraft missions is used in both cases. We present the kill-chain progression of each scenario and the Basic Performance Statistics comparison in Figures 5-21, 5-22, and Table 5-8.

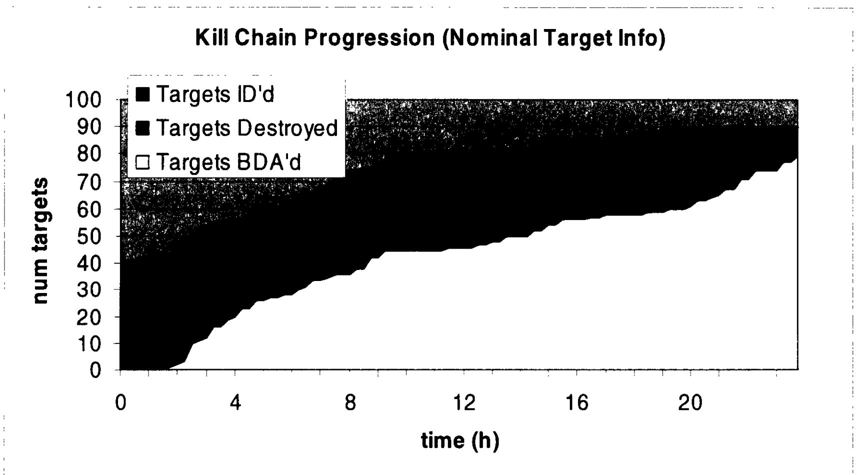

Figure 5-21: Kill-Chain Progression for Coordinated Scenario in Experiment 10

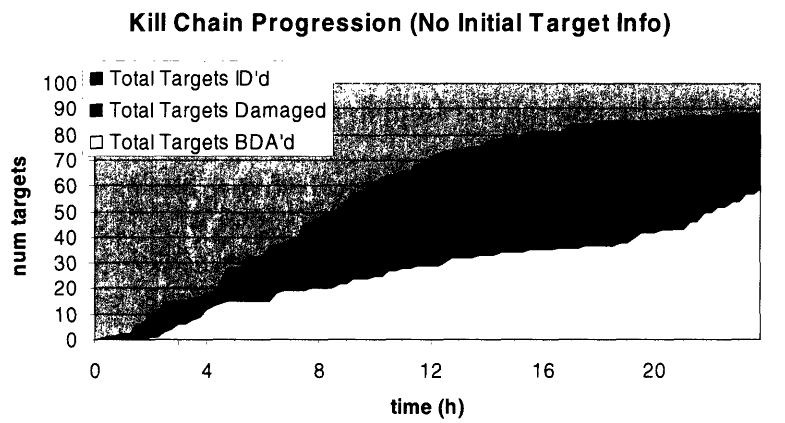

Figure 5-22: Kill-Chain Progression for IPB=0 case in Experiment 10

Basic Performance Statistics			
	IPB=1 (nominal)	**IPB=0 (none)**	
Number of Targets Destroyed Above Threshold	**90**	**86**	tgts out of 93
Targets First Hit by Type:			
Aircraft	**59**	**69**	tgts
EV	**32**	**18**	tgts
Aggregate Damage Fraction (ADF)	**0.921**	**0.818**	
Time to Achieve 0.85 ADF	**20.5**	**Never**	hours
Fraction High-Val Targets Destroyed	**1**	**0.875**	
Num Targets BDA'd	**79**	**59**	tgts
BDA'd by type:			
ISRtype2	**18**	**26**	tgts
Satellite	**61**	**33**	tgts
Avg Time from ID to Hit	**2.92**	**4.34**	hours
Avg Time from Hit to BDA	**4.72**	**5.41**	hours
Num AC Sorties	**619**	**604**	sorties
Num EV Sorties	**74**	**71**	sorties
Percent ADF per Sortie (incl. EV sorties)	**0.149%**	**0.135%**	dmg/sortie
Percent Attrition per Sortie	**2.423%**	**2.483%**	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**0.646%**	**0.662%**	attrit/sortie
Num Aircraft Attritions	**14**	**14**	
Attritions by type:			
Manned	**4**	**4**	attrit
Unmanned	**10**	**10**	attrit

Table 5-8: Basic Performance Statistics for Experiment 10

5.4.4 SIM Parameter Variations

- **Experiment 11: Increasing the Replan Interval**

  - **Discussion:** It is unclear how quickly future air and space planners will be able to plan and replan. Up to this point, we have assumed a replan time of 30 minutes, though whether this speed is achievable or not in the real-world could be questioned. The purpose of this test is to determine the effect of increasing the replan time from every 30 minutes to every 2 hours.

  - **Scenarios Involved:**

    A. Coordinated Scenario with ReplanInterval = 1800sec
    B. Coordinated Scenario with ReplanInterval = 7200sec

  - **Results:** As is seen in Figure 5-23, the ADFs of the two scenarios start to diverge around the sixth hour of the campaign. By the end of the campaign, Scenario A has an ADF of 0.921 while Scenario B has an ADF of only 0.687.

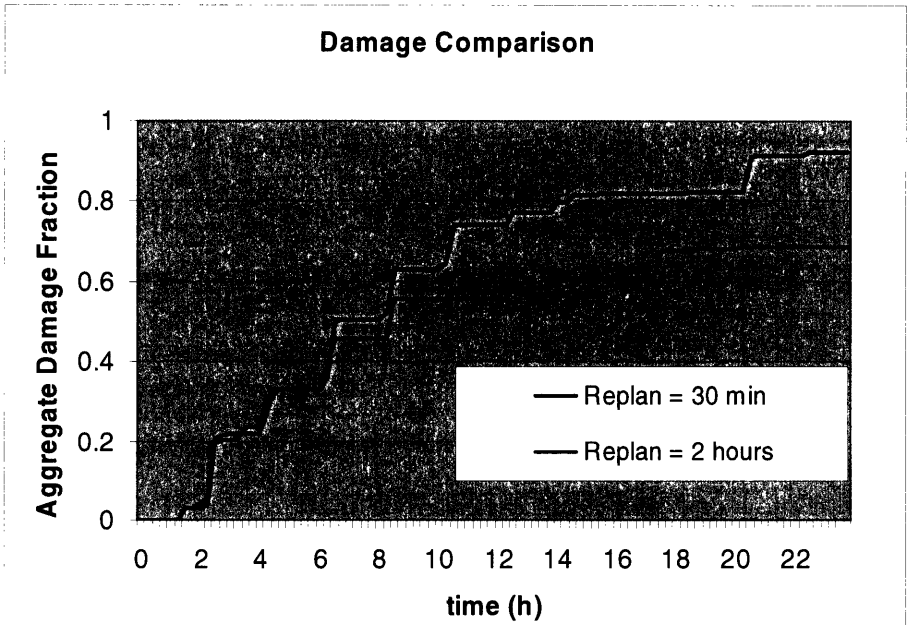

Figure 5-23: ADF Comparison for Experiment 11

- **Experiment 12: Addition of ORSC and SATC on Target Scenario 2**
  - **Discussion:** The attributes of Target Scenario 1 are somewhat purposefully contrived to exploit the obvious effects of adding resources and adding coordination. For example, the presence of a SAM barrier in Target Scenario 1 is favorable to a coordinated algorithm where ORSC can be used to destroy some of the barrier. It would be interesting to determine the benefit of adding resources and coordination in less of a contrived situation. The purpose of this test is to replicate Experiment 3 of section 5.3.3 on a generic target scenario.
  - **Scenarios Involved:**

    A. Base Scenario on Target Scenario 2 (given in Section 5.3.2)
    B. Base Scenario with ORSC and SATC on Target Scenario 2
    C. Coordinated Scenario on Target Scenario 2

    Results: In general, the same trends hold here as in Experiment 3. However, there are some notable exceptions. First, we compare Scenarios A and B. There are three fewer targets hit in Scenario B than in Scenario A. This might initially seem like an exceptional result considering Scenario B has 40 extra EVs at its disposal. It must be realized though, that both numbers of targets hit are near the max of 100 and also that Scenario B ends up with a much higher ADF (0.891 vs. 0.759). So while Scenario A does hit more targets, Scenario B gains more target "value." Also, Scenario B has a much higher manned attrition rate than Scenario A; one explanation is that there are more opportunities in Scenario B to attack SAMs – including attacking the same SAM multiple times if not completely destroyed on the first mission. All other comparisons between Scenario A and B are expected such as: 1) 36 more targets BDAd, and 2) reduction in average BDA time of 4.02 hours.

    Next, we compare Scenarios B and C. Although there are four more targets hit in Scenario C the ADF is actually slightly lower in Scenario C than in Scenario B. This is a result of ORS being used in more of a protective role. Every other metric shows improvement, including: 1) 18 more targets BDA'd, 2) a decrease in the average BDA time of 2.1

hours, and 3) a decrease in the manned aircraft attrition rate. Table 5-9 displays the Basic Performance Statistics for the experiment.

Basic Performance Statistics		**Target Scenario 2**		
	Just AC	**AC, ORSC, SATC**	**Coordinated**	
Number of Targets Destroyed Above Threshold	**97**	**94**	**98**	tgts out of 93
Targets First Hit by Type:				
Aircraft	**97**	**76**	**76**	tgts
EV	**0**	**18**	**22**	tgts
Aggregate Damage Fraction (ADF)	**0.759**	**0.891**	**0.879**	
Time to Achieve 0.85 ADF	**Never**	**14**	**14**	hours
Fraction High-Val Targets Destroyed	**0.775**	**0.9295**	**0.875**	
Num Targets BDA'd	**43**	**79**	**97**	tgts
BDA'd by type:				
ISRtype2	**43**	**32**	**42**	tgts
Satellite	**0**	**47**	**55**	tgts
Avg Time from ID to Hit	**3.77**	**3.10**	**2.57**	hours
Avg Time from Hit to BDA	**9.90**	**5.88**	**3.78**	hours
Num AC Sorties	**562**	**582**	**541**	sorties
Num EV Sorties	**0**	**78**	**75**	sorties
Percent ADF per Sortie (incl. EV sorties)	**0.133%**	**0.148%**	**0.139%**	dmg/sortie
Percent Attrition per Sortie	**2.669%**	**2.577%**	**2.773%**	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	**0.712%**	**1.890%**	**0.924%**	attrit/sortie
Num Aircraft Attritions	**12**	**19**	**13**	
Attritions by type:				
Manned	**4**	**11**	**5**	attrit
Unmanned	**8**	**8**	**8**	attrit

Table 5-9: Basic Performance Statistics for Experiment 12

- **Experiment 13: Addition of ORSC and SATC with Lower Probability of Attrition**

  - **Discussion:** The probability of aircraft attrition that is employed up to this point might be considered particularly high. This was partially done to exploit the capability of the coordinated algorithm to decrease the manned aircraft attrition rate. In actuality, such a high probability of attrition would correspond to an extremely (perhaps unrealistically) lethal air defense force. This test is used to investigate how the coordinated algorithm performs when the probability of aircraft attrition is lowered from 0.01 to 0.002. Note that this probability is conditioned upon an aircraft engagement with a SAM. An engagement with a SAM is itself probabilistic and defined in Section 5.2.1.

  - **Scenarios Involved:**

    A. Base Scenario with P(AttritGivenSAMengagement) = 0.002

B. Addition of ORSC and SATC with P(AttritGivenSAMengagement) = 0.002
C. Coordinated Scenario with P(AttritGivenSAMengagement) = 0.002

- **Results:** The Basic Performance Statistics comparison is shown in Table 5-10. It is evident that adding ORSC and SATC produces benefits in all areas except manned attrition rate. We attribute the increased attrition once again to the fact that there are more attacks made on SAMs in Scenario B. An apparently striking phenomenon is that Scenarios B and C appear to produce virtually the same results. While Scenario C does have a slightly higher ADF, it has one less target BDA, a slightly worse average BDA time, and a slightly worse manned attrition rate. One explanation is that the target scenario under these conditions is simply not discriminating enough. With such a low probability of attrition and with a target scenario where almost everything is destroyed, it is hard to distinguish between algorithms.

Basic Performance Statistics		Lower P(Attrition)		
	Just AC	AC, ORSC, SATC	Coordinated	
Number of Targets Destroyed Above Threshold	73	92	92	tgts out of 93
Targets First Hit by Type:				
Aircraft	73	76	75	tgts
EV	0	17	17	tgts
Aggregate Damage Fraction (ADF)	0.700	0.921	0.925	
Time to Achieve 0.85 ADF	Never	6.5	8.25	hours
Fraction High-Val Targets Destroyed	0.75	1	1	
Num Targets BDA'd	63	92	91	tgts
BDA'd by type:				
ISRtype2	63	64	63	tgts
Satellite	0	28	28	tgts
Avg Time from ID to Hit	1.95	1.92	2.41	hours
Avg Time from Hit to BDA	3.66	2.44	2.52	hours
Num AC Sorties	589	727	663	sorties
Num EV Sorties	0	71	73	sorties
Percent ADF per Sortie (incl. EV sorties)	0.119%	0.127%	0.139%	dmg/sortie
Percent Attrition per Sortie	2.547%	2.063%	2.262%	attrit/sortie
Percent Manned Aircraft Attrition per Sortie	0.000%	0.413%	0.452%	attrit/sortie
Num Aircraft Attritions	4	5	5	
Attritions by type:				
Manned	0	3	3	attrit
Unmanned	4	2	2	attrit

Table 5-10: Basic Performance Statistics for Experiment 13

[This Page Intentionally Left Blank]

6 Conclusions and Future Work

Effective military air and space operations revolve around the successful execution of the Targeting Cycle. Given sets of resources and targets, the goal of planners is to formulate missions on a recurring basis that best achieve the commander's objectives. As the set of resource types and mission scopes grows larger with new technology, coordinating among assets poses both challenges and potential benefits. The purpose of this thesis was to investigate the challenges of coordination and to quantify the potential benefits using the operational concept of Chapter 2. In this chapter, we summarize the work accomplished up to this point and offer suggestions for future work.

6.1 Summary

In the first five chapters, we presented an operational problem facing air and space planners, provided a functional framework for coordinating among three sub-controllers, formulated new algorithms for both the Coordination Controller and ORS Controller components, and integrated four pieces of software into one *CDASOCS* system. Using *CDASOCS*, it was shown that both the addition of, and coordination between, air and space resources can lead to significant improvements in the outcome of a battle campaign.

In Chapter 1, we gave an overview of the Targeting Cycle to familiarize the reader with the problem domain.

In Chapter 2, we set out to define the problem from an operational standpoint. A battle scenario including resource capabilities and target threat description was given. The target threat was comprised of high-value WMD facilities, a significant Integrated Air Defense force, and other conventional targets. The resources available involved aircraft, satellite, and ORS assets. Next, the command and control structure was discussed including day-to-day operations and certain coordination issues. Finally, we defined the Air and Space Operations Planning Problem, or *ASOPP*.

In Chapter 3, it was the goal to functionally model the Targeting Cycle in the context of *ASOPP*. First, modeling assumptions were presented. Then, a functional architecture of *ASOPP* was given involving a Coordination Controller, three sub-controllers, and a battlespace simulation. The two functional components developed for this thesis, CC and ORSC, were discussed in depth including the inputs, decisions, objectives, and constraints of each component. An outline of the remaining three (pre-existing) components was given.

In Chapter 4, we presented the *CDASOCS* system, showing the structural architecture of the system and giving new algorithms for CC and ORSC. In the first part of the chapter, we investigated previous research of problems similar to *ASOPP*. Then, we discussed certain constraints that were placed on *CDASOCS* as a result of the pre-existing software being used. Lastly, we formulated three new algorithms for CC and two new algorithms for ORSC, each offering different levels of capability.

In Chapter 5, the goal was to test the *CDASOCS* system and quantify its performance. First, we provided details on the software implementation and integration of the system. Then, we outlined specific parameter values that would be used in different scenarios, also defining a base scenario. Next, we performed a series of three main experiments that showed the benefit of 1) adding different sets of resources to the base version of just air operations and 2) coordinating among the different resources. Finally, we tested variations to the main parameter values, gleaning insight into the advantages and drawbacks of *CDASOCS* as well as the potential effects of making the same variations in an operational setting.

6.2 Future Work

Throughout the integration and testing of the *CDASOCS* system, it became apparent that certain areas of future research might make the system more realistic, lead to better solutions, and just prove interesting:

1. **Increased Functionality of Sub-controllers –** As suggested in Chapter 4, if the AC and SAT sub-controllers had the capability to accept time requests for each action, then CC would have a much-increased ability to coordinate among resources. Better coordination would be evidenced by a higher damage fraction, faster time to achieve 85% damage, lower attrition, etc. This is, perhaps, the most important first change to be implemented in *CDASOCS*.

2. **Increased Information Flow Between Functional Components –** In the current *CDASOCS* system, the ORS controller is the only sub-controller to provide CC with feedback. It is quite challenging to produce a collaborative system without quick feedback (although CC did get feedback from the SIM but only after the replan interval was over). If the sub-controllers could formulate their plans and then send them up to CC for review and possible change *before* their implementation, there would be *significant* reductions in information lag and the potential would exist for significant improvement to the coordinated system.

3. **Improved Estimation of Sub-controller Performance –** It is not the job of CC to do the work of Sub-controllers for them, although a better estimate of what they can do might lead toward an increased efficiency in terms of both sub-controller utilization and percentage of CC plans that are implemented. For instance, knowing that AC does not have the capability to process a certain target in the next planning interval might lead CC to assign the target to another sub-controller rather than AC.

4. **Formulation of CC as a Dynamic Program –** The function of CC is most realistically portrayed in the mathematical world as a probabilistic dynamic program. In CC as in Dynamic Programming, information becomes available in stages, decisions are

made in stages and make use of this information, and probabilities of future outcomes weigh heavily on the decisions. Moreover, DP is an optimal method and would provide the best possible solution over all policies.

5. Additional Realism/Functionality to ORS Sub-controller – It is currently assumed in *CDASOCS* that ORS assets can, at any time, strike any target that has been identified and that it takes a fixed amount of time for an LV/EV combination to reach the target. A more realistic system might take things into account such as launch windows, the effect of weather, target-movement, target-location, etc. on performance, and orbital mechanics to determine an estimated-time-on-target.

6. Additional Sub-controllers – There is always the possibility of modeling additional resource-types in the system to provide a more complete battle-scenario. Infantry and artillery units, underwater vehicles, aircraft carriers, refueling aircraft, close-air-support aircraft, communications and navigation satellites, etc. would each add an element of completeness toward quantifying the benefit of coordination among resources.

Appendix A: Glossary of Acronyms

AC.................................... Aircraft Controller

ADF.................................. Aggregate Damage Fraction

ASOPP............................. Air and Space Operations Planning Problem

AOC................................. Air Operations Center

ATO................................. Air Tasking Order

BDA................................. Bomb Damage Assessment

C3..................................... Command and Control Facility

CAV................................. Common Aero Vehicle

CC.................................... Coordination Controller

CDASOCS........................ Coordinated Dynamic Air and Space Operations Control System

COG................................. Center of Gravity

CONOPS........................... Concept of Operations

DAP.................................. Dynamic Assignment Problem

DP..................................... Dynamic Programming

EPOS................................ Earth Phenomena Observing System

EV.................................... Entry Vehicle

FCASOS........................... Functional Coordinated Air and Space Operations System

FOV.................................. Field of View

FWE................................. Fighter Wing Equivalent

HASOC............................ Heuristic Air and Space Operations Coordinator

HORSC............................ Heuristic Operationally Responsive Spacelift Controller

IADS................................ Integrated Air Defense System

IBMS................................ Integrated Battle Management System

ID..................................... Target Identification

IPB................................... Intelligence Preparation of the Battlefield

ISR................................... Intelligence Surveillance and Reconnaissance

IP...................................... Integer Programming

IR...................................... Infrared

JAOC................................Joint Air Operations Center
JAOP................................Joint Air Operations Plan
JASTC..............................Joint Air and Space Tasking Cycle
JDAM...............................Joint Direct Attack Munitions
JFACC..............................Joint Forces Air Component Commander
JFC...................................Joint Forces Commander
JSOW...............................Joint Standoff Weapon
LAN.................................Local Area Network
LEO..................................Low Earth Orbit
LGB..................................Laser Guided Bomb
LP.....................................Linear Programming
LV....................................Launch Vehicle
MEA.................................Munitions Effectiveness Assessment
MIP...................................Mixed Integer Program
MOE.................................Measures of Effectiveness
NASOC............................Nominal Air and Space Operations Coordinator
OASOC............................Optimizing Air and Space Operations Coordinator
OORSC............................Optimizing Operationally Responsive Spacelift Controller
OPCON............................Operational Control
ORS..................................Operationally Responsive Spacelift
ORSC...............................Operationally Responsive Spacelift Controller
PMF..................................Probability Mass Function
POMDP............................Partially Observable Markov Decision Process
RAAN..............................Right Ascension of the Ascending Node
SAM.................................Surface to Air Missile
SATC...............................Satellite Controller
SIM..................................Battlespace Simulation
SOV..................................Space Operations Vehicle
SP.....................................Stochastic Programming
STRK...............................Target Strike
TCT ....Time-Critical Target (also Time-Sensitive Target)

TNL.................................... Target Nomination List

TST.................................... Time-Sensitive Target (also Time-Critical Target)

UAV.................................. Unmanned Aerial Vehicle

WMD................................ Weapons of Mass Destruction

WTA.................................. Weapon Target Assignment

References

[1] Abramson, Mark, et al., "The Design and Implementation of Draper's Earth Phenomena Observing System (EPOS)." *Proceedings of the AIAA Space 2001 Conference,* Albuquerque, NM, August 2001.

[2] Adams, Milton. *Joint Air-To-Ground, Unified Adaptive Replanning: JAGUAR.* Charles Stark Draper Laboratory Report R-2959. Cambridge, Massachusetts. March, 2004.

[3] "Airpower Overview." *Federation of American Scientists.* <http://www.fas.org/man/dod-101/sys/ac/overview.htm> (June 13, 2004)

[4] Barth, Christopher D. *Composite Mission Variable Formulation For Real-Time Mission Planning*, Master's Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2001.

[5] Bertsimas, Dimitris, and Sim, Melvyn. "The Price of Robustness," *Operations Research*, Vol. 52, pp. 35-53, 2004.

[6] Bertsimas, Dimitris, and Sim, Melvyn. "Robust Discrete Optimization and Network Flows". *Mathematical Programming,* Series B, 98:49-71, 2003

[7] Birge, John R., and Louveaux, Francois. *Introduction to Stochastic Programming.* New York: Springer-Verlag New York, Inc., 1997.

[8] Casella, George and Roger L. Berger. *Statistical Inference.* 2nd Edition. Duxbury Thomson Learning, 2002.

[9] David, Leonard. "Air Force to Focus on Quick Reaction Spacelift Vehicles." *Space.com.* March 28, 2003. <http://www.space.com/businesstechnology/technology/af_space_030328.html> (March 04, 2004).

[10] Henderson, Scott, Lt Col., USAF. National Defense Fellow. MIT Security Studies Program. Personal Interview. March 04, 2005.

[11] Joint Publication 3-0. *Doctrine for Joint Operations.* September 10, 2001.

[12] Joint Publication 3-14. *Joint Doctrine for Space Operations.* August 09, 2002.

[13] Joint Publication 3-30. *Command and Control for Joint Air Operations.* June 05, 2003.

[14] Joint Publication 3-60. *Joint Doctrine for Targeting.* January 17, 2002.

[15] Kelly, Ricky B. *Centralized Control of Space: The Use of Space Forces by a Joint Force Commander.* School of Advanced Airpower Studies, Air University, Maxwell Air Force Base, Alabama, 1994.

[16] Linder, Craig. "Microsatellites to Mimic Migrating Birds." *Space.com.* July 09, 2000. <http://www.space.com/businesstechnology/technology/microsat_formation_000707.html> (March 04, 2004).

[17] Moran, Michael J. "An Evolving Doctrine: Force Application from Space." Air Command and Staff College, Air University, Maxwell Air Force Base, Alabama, March 22, 2000.

[18] Murphey, Robert A., "An Approximate Algorithm For a Weapon Target Assignment Stochastic Program." *Air Force Research Laboratory, Munitions Directorate.* Eglin AFB, Florida, 1999.

[19] Phillips, Terry, and Bob O'Leary. "Common Aero Vehicle (CAV) on Orbit." Schafer Corporation. September 06, 2003. <http://www.dtic.mil/matris/sbir/sbir041/srch/af031d.doc>. (February 21, 2005).

[20] Pincus, Walter. "Pentagon Has Far-Reaching Defense Spacecraft in Works; Bush Administration Looking to Space to Fight Threats." *The Washington Post.* Washingon, D.C., March 16, 2005.

[21] Singer, Jeremy. "U.S. Air Force Sees Quick and Frequent Launches in its Future." *Space News Online.* July 14, 2003. <http://www.space.com/spacenews/spacenews_businessmonday_030714a.html> (March 04, 2004).

[22] Sills, Larry G. "Space-Based Global Strike: Understanding Strategic and Military Implications." Occasional Paper No. 24, Center for Strategy and Technology, Air War College, Maxwell Air Force Base, Alabama, August 2001.

[23] Sirak, Michael. "US Air Force Mulls New Space Launcher." *Jane's Defense Weekly.* Washington DC, October 29, 2003.

[24] Spivey, Michael Z. and Powell, Warren B. *The Dynamic Assignment Problem.* Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey, 2003.

[25] Stiles, Brian A. *Application of Sequential Auction Techniques to Nonlinear Targeting Assignment for Space-Delivered Entry Vehicles,* Master's Thesis, Rice University, Houston, Texas, 2004.

[26] Yost, Kirk A. *Solution of Large-Scale Allocation Problems with Partially Observable Outcomes.* Ph.D. Dissertation in Operations Research. Naval Postgraduate School, Monterey, California, September 1998.

[27] Zarybnisky, Eric J. *Allocation of Air Resources Against an Intelligent Adversary,* Master's Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2003.