NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZING THE AIR-TO-GROUND KILL CHAIN FOR TIME-SENSITIVE TARGETS

by

Bradley A. Bloye

September 2009

Thesis Advisor:	W. Matthew Carlyle
Second Reader:	Gerald Brown

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Optimizing the Air-to-Ground Kill Chain for Time-Sensitive Targets	**5. FUNDING NUMBERS**
6. AUTHOR(S) Bradley A. Bloye	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

When groups of platforms, sensors, and weapons are able to communicate with each other in real-time, they form a network. Modern warfare increasingly involves network-centric operations, the military strategy that seeks to translate informational advantages gained through the cooperation of all platforms in the network into increased overall mission effectiveness. For this thesis, the Time-to-Kill is our metric to quantify mission effectiveness because a given time-sensitive target is vulnerable to attack only for a very short time. This thesis develops an optimizing heuristic kill chain assessment tool, "KCAT," that (a) rapidly identifies capability gaps and (b) generates guaranteed feasible schedules that minimize the time-to-kill for a given air-to-ground strike scenario. KCAT allows warfare analysts, budget programmers, and mission planners to quantitatively examine the value of network-centric warfare in time-sensitive targeting scenarios. In addition to optimizing existing platform and weapon network effectiveness, KCAT allows experimentation with future concepts and capabilities that are important for informing procurement and training decisions.

14. SUBJECT TERMS Kill chain, air-to-ground, optimization, makespan, time-sensitive targeting.	**15. NUMBER OF PAGES** 75
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

OPTIMIZING THE AIR-TO-GROUND KILL CHAIN FOR TIME-SENSITIVE TARGETS

Bradley A. Bloye
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author:	Brad Bloye
Approved by:	W. Matthew Carlyle Thesis Advisor
	Gerald G. Brown Second Reader
	Robert F. Dell Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

When groups of platforms, sensors, and weapons are able to communicate with each other in real-time, they form a network. Modern warfare increasingly involves network-centric operations, the military strategy that seeks to translate informational advantages gained through the cooperation of all platforms in the network into increased overall mission effectiveness. For this thesis, the Time-to-Kill is our metric to quantify mission effectiveness because a given time-sensitive target is vulnerable to attack only for a very short time. This thesis develops an optimizing heuristic kill chain assessment tool, “KCAT,” that (a) rapidly identifies capability gaps and (b) generates guaranteed feasible schedules that minimize the time-to-kill for a given air-to-ground strike scenario. KCAT allows warfare analysts, budget programmers, and mission planners to quantitatively examine the value of network-centric warfare in time-sensitive targeting scenarios. In addition to optimizing existing platform and weapon network effectiveness, KCAT allows experimentation with future concepts and capabilities that are important for informing procurement and training decisions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

We wish to develop the capability to destroy time-sensitive targets as quickly as possible, and we wish to ascertain the technical capabilities that are critical to this objective.

The threat we face today from theater ballistic missiles (TBMs) is enormous and growing larger. Numerous countries of interest possess TBMs, and some pursue their development as delivery vehicles for weapons of mass destruction. When precise intelligence on the transformer-erector launchers (TELs) that launch TBMs becomes available, we must have the capability to destroy them quickly. This capability is vital to our ability to secure and dominate the battle space. Additionally, we need to identify the areas in which we lack the technical means to accomplish rapid time-sensitive strike.

The Warfare Analysis and Integration Branch of the Naval Air Systems Command (NAVAIR), located at China Lake Naval Air Weapons Station, CA, conducts analysis of alternatives studies for platform and weapon systems, as well as sensitivity analyses for Concepts of Operations (CONOPS). The air-to-ground strike kill chain, the sequence of events that must occur successfully in correct sequence in order for a strike aircraft to destroy a ground target, is at the heart of these analyses. Current analytical methods use simulation and manual analysis. These analyses address individual target kill chains vice multiple kill chains occurring simultaneously.

We develop a kill chain assessment tool (KCAT) that builds feasible schedules for multiple simultaneous kill chains. KCAT run times on a 3.16 GHz Dell Quad-core desktop computer at the Naval Postgraduate School have never exceeded two seconds for a scenario involving as many as 45 targets and 12 platforms. KCAT uses a Microsoft Excel interface and produces output in Microsoft Excel. KCAT can thus be rapidly distributed and reliably operated in a variety of computing environments without the expense and difficulty of software licenses or special training. In particular, KCAT can run on Navy-Marine Corps Intranet Computers (NMCI).

KCAT produces feasible solutions for multiple simultaneous kill chains. KCAT performs well under the assumptions we have stated for a wide variety of scenario starting configurations and associated constraints. Heuristic improvements significantly reduce the maximum time-to-kill for many scenarios, and KCAT results look face-valid. KCAT outputs allow the planner to analyze air-to-ground attack scenarios quickly and effectively. These analyses inform larger questions related to platform and weapon capabilities, limitations, and potential.

ACKNOWLEDGMENTS

I would like to thank the many people who have assisted me in the writing of this thesis:

Professor Matt Carlyle, thank you for guiding me through the thesis process and for spending the time and attention to help create a quality product. The journey has been challenging at times, but certainly rewarding and pleasant throughout. I have learned a tremendous amount from you; I am very grateful for that.

Professor Gerald Brown, thank you for lending your distinguished experience and knowledge to this project. I learned an enormous amount in a very short period of time thanks to you.

Ken Amster, thank you for your personal support and guidance, and for making the effort to promote the value of NPS - China Lake collaboration.

Bill Brickner, thank you for the opportunity to work with your branch and to contribute to an interesting and important area of study. Thank you for your many ideas and for the time you spent to guide me through this project. I very much appreciate your support and your warm welcome at China Lake.

I would also like to thank a number of truly outstanding professions with whom I had the pleasure and honor of working with at China Lake:

Jim DeSanti, Ricky Fielding, Bob Smith, and Bruce Fecht, thank you for your inspiration and assistance.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PURPOSE AND OVERVIEW

The Warfare Analysis and Integration Branch of the Naval Air Systems Command (NAVAIR), located at China Lake Naval Air Weapons Station, CA, conducts analysis-of-alternatives studies for platform and weapon systems, as well as sensitivity analyses for Concepts of Operations (CONOPS). The air-to-ground strike kill chain, the sequence of events that must occur successfully in correct sequence in order for a strike aircraft to destroy a ground target, is at the heart of these analyses. The lack of a flexible, quick, and accurate network-centric kill chain model significantly hampers the ability of NAVAIR to produce analyses rapidly. This thesis develops an analysis tool that produces feasible schedules for multiple simultaneous kill chains: the Kill Chain Assessment Tool (KCAT).

We use an unclassified scenario called “Ithaca” as a case study. We break down or combine the most frequently used phases of the kill chain, Find–Fix–Target–Track–Engage–Assess (F2T2EA), into functional areas that are specific to the air-to-ground strike mission: Detect–Track–Identify–Approve–Launch-Control-Assess. KCAT assigns platforms and weapons to targets, and produces feasible schedules that assign all phases of the kill chain to scenario targets.

We develop a heuristic to optimize air-to-ground kill chains, rather than using commercial mathematical-programming software. KCAT can thus be rapidly distributed and reliably operated in a variety of computing environments without the expense and difficulty of software licenses or special training. In particular, KCAT can run on Navy-Marine Corps Intranet Computers (NMCI).

B. BACKGROUND

1. Problem Statement

Kill Chain analysis is not new. Researchers have studied kill chains for years; the resulting analyses have identified important current and future capability gaps. These capability gaps, once identified, become critical acquisition issues for the Navy and the Department of Defense. While recent analyses of kill chains have successfully identified capability gaps, to date *there have been no kill chain models that support planning multiple simultaneous kill chains against multiple targets*. The primary contribution of this thesis is the development of an analysis tool that accomplishes the feasible construction of such kill chains. Future research can build upon this work to produce provably optimal solutions.

KCAT is the first decision support tool that explicitly models the time-to-kill for several kill chains at once. KCAT represents a significant step beyond NAVAIR's current analysis techniques that only identify capability gaps in single kill chains. Furthermore, KCAT is *prescriptive,* in contrast to the existing *descriptive* analyses.

We envision using KCAT to analyze missions associated with a major contingency operation (MCO), and drawing tactical, operational, or strategic insights from the results. These insights can inform current CONOPS planning, and may suggest new alternative solutions. KCAT allows the planner to evaluate future platform or weapon systems within the context of their proposed mission. We assess the relative importance of the individual phases of the kill chain by examining scenarios that attack time-sensitive targets.

2. Definitions and Terms

We use the following terms throughout this thesis:

- Kill Chain. The sequence of events that must succeed to destroy a target. The most common usage of the term includes Find, Fix, Track, Target, Engage, and Assess (F2T2EA). This thesis tailors the generic kill chain to specifically address air-to-ground strikes:

  - *Detect* – perceive an object of possible military interest
  - *Track* – precisely and continuously find targets by radar, visual, or other means
  - *Identify* – correlate a contact of interest (COI) with a desired target via visual, forward-looking infrared, radar, or other means sufficiently well to release ordnance
  - *Approve* – a higher authority gives permission to commence attack
  - *Launch* – a weapon is released from its host platform
  - *Control* – providing the information necessary to a weapon after launch
  - *Assess* – determining effects of attacks on targets
- Platform. U.S. or coalition military aircraft that can be used in air-to-ground strike.
- Weapon. Air-to-ground ordnance that scenario platforms may carry.
- Target. The object of the attack.
- Pd. The probability of a weapon successfully damaging a target.
- Pk. The probability of a weapon successfully destroying a target.
- Operational range. The range at which a weapon, when released, is expected to achieve a Pd of at least 0.7; in other words, the desired release distance from the target.
- Maximum weapon kinematic range. The maximum distance that a weapon can travel.
- Maximum weapon acquisition range. The maximum distance at which a weapon can acquire a target with an on-board sensor.
- Weapon detection range. The range at which a weapon is first capable of detecting its intended target.

- Weapon minimum range. The minimum distance from a target that a weapon can be released and still successfully reach the target.
- Time-sensitive target. A target with a limited period of vulnerability to attack, or a target that will become a serious threat within a short period of time.
- Strike package. The group of platforms working together to achieve an air-to-ground strike.

C. HISTORY OF KILL CHAIN ANALYSIS

Kill chain analysis is not new. There are two major Navy programs focused on kill chain analysis: (1) The Program Manager, Precision Strike Weapons (PMA 201) Kill Chain Analysis Process (KCAP 201) of 2006 (Naval Air Systems Command, PMA-201, 2006) and (2) Horizontal Integration and Capability Based Assessment Program (HICAP), which is ongoing at NAVAIR (Smith, 2008).

The purpose of the PMA 201 Kill Chain Analysis is to look for areas in the kill chain that are broken or weak, within the context of specific missions. The primary areas of interest are land-moving target missions, time-critical target missions, and air-to-air missions. In its initial analysis, KCAP 201 assessed 46 different kill chains using data and subject-matter expertise from 14 NAVAIR program management offices. KCAP 201 used Microsoft Excel and an activity-based version of System Architect, a Department of Defense “Architecture Framework” tool owned and marketed by Telelogic (IBM, 2009), to perform kill chain scoring and to generate kill chain outputs. KCAP 201 successfully identified several broken kill chains, generated associated issue sheets (a standard form for troubleshooting capability gaps that usually leads to procurement), and in some cases was able to obtain funding approval to implement required changes (Naval Air Systems Command, PMA-201, 2006).

HICAP is an ongoing NAVAIR project that seeks to increase the degree of integrated warfighting capability and interoperability from the resources available. It aims to re-focus the Navy procurement system on desired mission end-states rather than on individual capabilities that may or may not contribute to the end-states. HICAP accomplishes this through a complex integration process involving CONOPS working groups, program management offices, Fleet inputs, threat assessments, and kill chain analyses. In HICAP, for a given kill chain, analysts manually apply stoplight scoring (i.e., grading as “red,” “yellow,” or “green,” based on standardized thresholds) to each step of a kill chain for individual scenarios. HICAP then integrates kill chain analyses with threat assessments over a time horizon. The result is an answer to the question “In the context of the Kill Chain Scenario and Fleet approved CONOPS, does the capability exist to effectively employ the weapon against the target of interest?” The answers to this question for various scenarios then inform decisions about procurement (Smith, 2008).

D. SCOPE AND LIMITATIONS

Kill chains describe any sequence of events resulting in a successful attack. Here, we address kill chains for air-to-ground strikes of time-sensitive mobile transformer-erecter launchers (TEL) targets. For our baseline scenario, we limit our analysis to four platforms and 12 weapons, and explore variations in number of targets, number of platforms, and weapon types. KCAT has the capability to examine many other types of scenarios as well.

This thesis simplifies three of the kill chain phases for modeling purposes. For example, the “Approval” phase consists of a very complex and dispersed network involving multiple platforms, communications networks, and command and control facilities. Thus, the time required for attack approval varies greatly, and depends upon the particular circumstances of the desired attack. We likewise simplify the “Detect” and “Identify” phases so that we can focus on the development of the overall optimization algorithm.

E. THESIS ORGANIZATION

Chapter II begins with an explanation of Ithaca, the fictitious region in which kill chain scenarios are constructed. We then present a brief overview of the theater ballistic missile (TBM) threat and time-sensitive strike (TST). Lastly, we discuss the air-to-ground kill chain in detail. Chapter III describes our air-to-ground strike kill chain optimization model and the heuristic algorithm we invented to solve it. Chapter IV presents heuristic solutions. Chapter V provides conclusions and areas for future work.

II. THE AIR-TO-GROUND STRIKE KILL CHAIN FOR TIME-SENSITIVE TARGETS

A. ITHACA

1. Background

Because this thesis is unclassified, an imaginary location called “Ithaca” will serve as the geographic setting for KCAT scenarios. The Warfare Analysis and Integration Branch of NAVAIR derived Ithaca from the USWESTCOM Joint Intelligence Center scenario “Pacifica” of 2003. Adjustments from the original Pacifica scenario include changing country names, revising the socio-political background, and updating the order of battle (Loibl, 2008).

Two joint U.S.-UK projects have utilized the Ithaca scenario. Ithaca’s focus is on naval air forces; it currently does not include land, surveillance assets, or logistics forces. NAVAIR and other organizations use Ithaca for various warfare analyses.

This thesis uses the portions of Ithaca related to the tactical ballistic missile (TBM) threat. Although Ithaca is fictitious, the scenario mirrors the reality of several real-world situations very closely, and thus the analytical results obtained using Ithaca are quite realistic. We use unclassified data throughout this thesis in terms of TBM dwell times, weapon capabilities, etc.; however, the planner can easily modify these parameters to match reality.

2. Geographic Description

Ithaca is an island that incorporates the states of California, Oregon, Nevada, and Arizona, but shifted from actual latitudes and longitudes by -10 degrees and -80 degrees, respectively. This places Ithaca in the Pacific Ocean between Hawaii and Guam. The approximate position of the center of Ithaca’s land mass is 20° N, 165° E.

Three fictitious nations share the island of Ithaca: Trinacria (California), Scheria (Nevada and Arizona), and Cythera (Oregon). Trinacria is an aggressor nation that attacks Scheria in order to take full control of shared natural resources. When Scheria appeals to the international community for security assistance, Trinacria threatens to attack coalition forces coming to Scheria's aid; Trinacria repositions its ballistic and cruise missile launchers accordingly.

Trinacria has a robust integrated air defense system (IADS). The Trinacria IADS includes 2nd, 3rd, 4th, and 5th generation fighter interceptors, strategic surface-to-air missile (SAM) batteries, anti-air artillery (AAA) batteries, and early warning radar. One air defense operations center maintains situational awareness of air operations. Trinacria bases its defensive doctrine on multiple layers of defense: fighter interceptors, SAMs, and AAA (Loibl, 2008).

3. Trinacrian Theater Ballistic Missile (TBM) Threat

Trinacria poses a robust ballistic missile threat. Table 1 describes the three different types of Trinacrian ballistic missile launchers and missiles.

System	Status	Range	Missiles	CW Warheads	Launchers	Reloads/TEL
SRBM Type 1	OP	400 km	294	None	10	29.4
SRBM Type 2	OP	700 km	295	Possible	15	19.6
MRBM	OP	2250 km	121	Possible	25	4.8

Table 1. Trinacria Ballistic Missile Threat. From left to right, the SRBM Type 2 missile system is operational with a range of 700 km. 295 such missiles are in stock, and each is capable of carrying a chemical warhead. There are 15 SRBM Type 2 launchers in Trinacria, and they each have an average of 19.6 missile reloads per TEL.

Trinacria can launch its TBMs from any of three different launch areas. Table 2 describes them.

Area	Range	SRBM Type 1 TELs	SRBM Type 2 TELs	MRBM TELs
TBM A-1	700 km	0	5	12
TBM A-2	400 km	10	0	0
TBM A-3	700 km	0	10	13

Table 2. Trinacrian TBM launch areas. From left to right, the maximum range of the types of TELs operating within TBM Area 1 is 700 km, the number of SRBM Type 1 TELs in TBM Area 1 is zero, the number of SRBM Type 2 TELs in TBM Area 1 is five, and the number of MRBM TELs in Area 1 is 12.

Figure 1 depicts these launch areas.

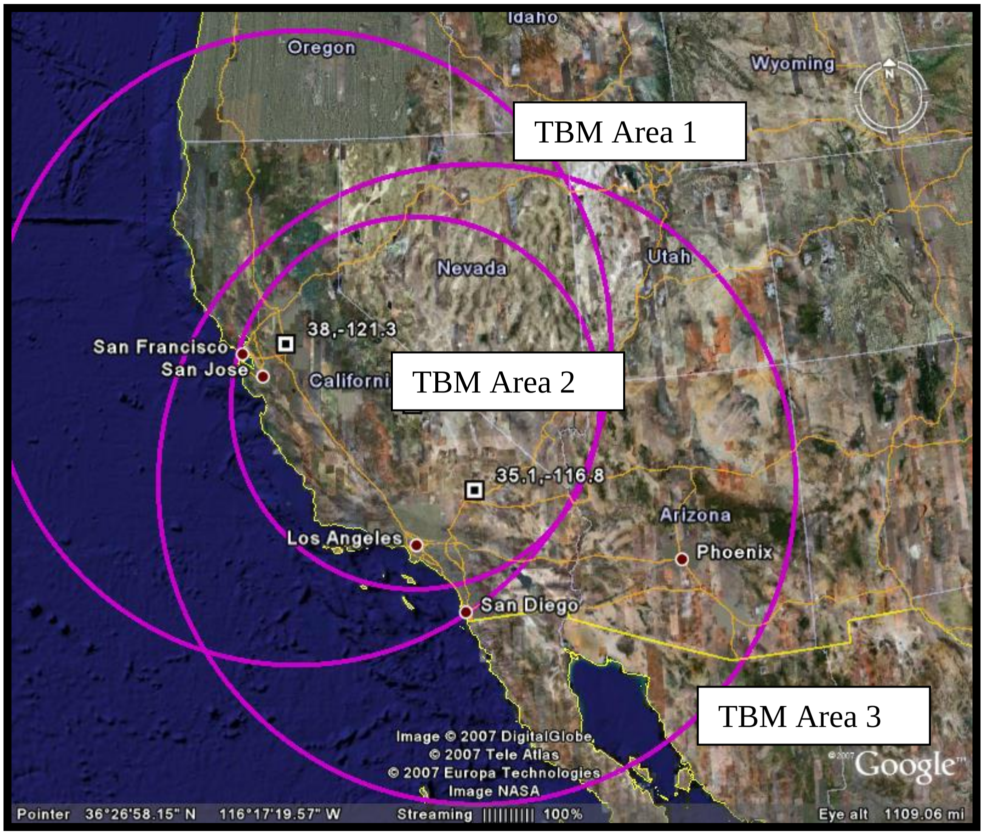


Figure 1. Map of Trinacrian TBM launch areas (From: Google Maps). Each circle represents the maximum range for the launch area as described in Table 2.

4. Trinacrian TBM CONOPS and Timeline

We base the Trinacrian TBM CONOPS and timeline of this thesis on a scenario presented by an earlier thesis that analyzed the kill chain for time-critical strike (Brickner, 2005).

Within each TBM Area, there are numerous sites for transformer-erector launchers (TELs) to either hide or launch missiles. Trinacrian TBM CONOPS call for two launch waves per day: the first wave launches from 0600–1000, and the second from 1800–2200.

TBMs depart from random hidden sites on command, in order to achieve all launches for that wave within a 25-minute epoch. The launch site, relative to the hidden site, must be between 4.3 and 43 nm away. We compute the travel time of each TBM assuming a straight-line distance and an average speed of 21.5 mph. After arrival to the launch site, TBM launch occurs in 14–22 minutes for SRBMs, and 25–35 minutes for MRBMs. After launch, TBMs depart the launch site for a new hide site in 1–3 minutes for SRBMs and 3–6 minutes for MRBMs. The new hide site, once again, must be between 4.3 and 43nm away (Brickner, 2005).

B. THE THEATER BALLISTIC MISSILE CHALLENGE

1. Background

Ballistic missiles pose a serious and growing threat to U.S. and coalition forces overseas, and increasingly, to the continental United States. Many countries view ballistic and cruise missile systems as cost-effective weapons, and as symbols of national power. They present an asymmetric threat to U.S. airpower, but most alarming of all they can carry weapons of mass destruction over long distances.

Ballistic missiles are attractive to many nations because they are relatively inexpensive but effective even against an opponent with a robust air defense network. Ballistic missiles enable attacks where attacks with manned aircraft would be impractical or prohibitively expensive (National Air and Space Intelligence Center, 2009).

Ballistic missiles are currently in widespread development. Many organizations expect their number and variety to increase in the near future. They are useful as deterrent or coercive weapons, and have fewer maintenance, training, and logistic requirements than manned aircraft. Mobile transformer-erector launchers (TELs) are very survivable because they can be hidden easily, thus making them difficult to attack.

Warring nations have used ballistic missiles in several conflicts over the last 25 years, including the Iran–Iraq war, the Afghan civil war, the 1991 and 2003 Persian Gulf conflicts, and the Russian military action in Chechnya and Georgia.

The ballistic missile threat continues to increase as missile technology proliferates throughout the world. Over 20 countries currently have ballistic missile systems; thus, almost assuredly, ballistic missiles will be involved in future conflicts involving U.S. forces. (Feickert, 2005).

2. Theater Ballistic Missile Defense (TBMD)

There are two components of theater ballistic missile defense (TBMD): counterforce and active defense. Counterforce refers to destroying the missile launcher before it successfully achieves launch, i.e., pre-launch, whereas active defense refers to attacking the missile after it has been launched, i.e., post-launch.

Most current TBMD plans center around the active defense strategy. Practitioners of active defense seek to exploit the vulnerabilities that a missile in flight presents at various phases of its trajectory. A ballistic missile in flight is certainly difficult to attack, but active defense considers it easier to detect and attack such a missile than a hidden, mobile launcher. Brown et al. (2005) develop an optimization model that plans the pre-positioning of ballistic missile defense platforms to minimize the worst-case damage an attacker can achieve. Combatant commanders use this and other models to make the best use of TBMD platforms engaged in an active defense strategy.

Combatant commanders as well as academics have traditionally considered counterforce strategies more difficult to perform than active defense strategies. Adversaries can hide TELs easily; SRBM and MRBM launchers are mobile and can launch a missile within minutes of arriving at a launch site. This makes it very difficult to attack such a launcher in time to prevent a successful launch. Threat nations can hide and harden their intermediate range ballistic missile (IRBM) and ICBM missile launch facilities against attack with relative ease, once again making the counterforce strategy very challenging.

Marshall (1994) draws an analogy between anti-submarine warfare (ASW) and theater ballistic missile defense (TBMD). He indicates that both missions require very similar actions in sequence: searching, detecting, localizing, classifying, and attacking the contact of interest. Marshall applies the lessons learned over many years of ASW to TBMD: “Notice that ASW was never referred to as torpedo defense. Attempts were not made to kill the torpedo in the water; efforts were always concentrated on going after the launcher (the submarine) or the infrastructure necessary for it to operate.”

He develops a mathematical model that analyzes the effects of counterforce and active defense strategies, and states his results as follows: “It is shown that without counterforce an active defense system could require an impractical number of weapons to counter incoming missiles and/or their warheads. This number is shown to decrease geometrically as effective counterforce is used, so that the expected number of warheads killed increases dramatically with counterforce that is only modestly effective.”

He concludes his analysis by stating that both counterforce and active defense should be essential elements in any future TBMD system:

> *Without counterforce it will be relatively easy for the enemy to overwhelm a feasible active defense system. A system that can successfully destroy launchers and their crews will provide considerable leverage in reducing the numbers of active defense weapons required; this leverage increases dramatically as the number of warheads on each missile increases... the successful counterforce against launchers on land will require efforts in cueing, search, detection, localization, classification, and destruction. Current efforts can be thought of as attempting to skip from cueing (for example, framing datum information after launch) to attack. Future reports will consider how one might best accomplish the in-between phases to produce successful counterforce against mobile missile launchers.*

One objective of this thesis is to provide a mechanism by which analysts can examine such counterforce strategies, in light of newly available technologies related to stealth, detection, and high-speed weapons.

C. TIME-SENSITIVE STRIKE CONSIDERATIONS IN TBMD

1. Background

Time sensitive conventional strike from long standoff ranges into restricted or denied territory has been an operational, policy, and acquisition challenge for a long time. This difficult mission has appeared in many studies and reports as a hard problem for which no satisfactory solution appears readily available. In situations where time is not a factor, or where the U.S. has sufficient forces deployed nearby, the U.S. has demonstrated its ability to strike at identified threats effectively (Defense Science Board, 2009).

According to the Defense Science Board (DSB), "...the current DoD focus on delivery platforms for Time Critical Strike needs to be balanced with a considerably increased focus on ISR, munitions, C3, SOF, and exercises." The DSB found that "Covert, loitering strike systems enabled by robust target ISR and tracking, C3 and fire control capabilities would revolutionize global strike for both the long war and for deterrence of rogue and near-peer nations."

Regarding SCUD usage during Operation DESERT STORM, General Schwarzkopf stated, "The (SCUD) launchers turned out to be more elusive than we'd expected. We picked off a few, but just as often bombers would streak to a site where a missile had been launched only to find empty desert."

Wilson (2002) maintains that the realization of the U.S. military's shortfalls in its ability to attack fleeting, mobile targets has resulted in an increased emphasis on time-critical targeting by both military and civilian leadership. Time-sensitive targets will most likely continue to present the Joint Force Commander (JFC) with a significant challenge in future conflicts. In order to engage targets in single-digit minutes by the year 2015, a combination of hypersonic missiles and unmanned combat air vehicles, utilizing Advanced Targeting Recognition and the global information grid, will be necessary (Wilson, 2002).

D. ELEMENTS OF THE AIR-TO-GROUND STRIKE KILL CHAIN

1. Generic Air-to-Ground Strike Kill Chain Model

The term “kill chain” is general in nature, and can refer to any situation in which a sequence of events is required to destroy a target. For air-to-ground strike, kill chains are sequences of events that must each succeed in order for an air-to-ground attack to be successful.

Kill chains focus on the target, not a particular attacking platform. The phases of the kill chain are not things that a platform must do to attack a target with success, but instead things that some combination of platforms for each target must achieve.

The most commonly used form of the kill chain is “F2T2EA,” or Find–Fix–Track–Target–Engage–Assess. In this model, friendly platforms work together to find targets, and then they fix the target’s location with enough precision to enable an engagement. Next, they track the target; eventually, friendly platforms pass the target’s location information to a platform carrying a weapon for launch. This platform launches the weapon, and then the platform supports the weapon in-flight, as required, until impact. Finally, a friendly platform assesses the target area in order to determine whether the attack was successful.

2. KCAT Kill Chain

In this thesis, we take the generic kill chain model and modify it based on operational activities in order to capture more accurately he essential elements of air-to-ground strike with time criticality. Our modified sequence is Detect–Track–Identify–Approve–Launch–Control–Assess.

The first phase of KCAT’s kill chain model is “Detect.” In this phase, platforms attempt to detect contacts of interest (COI). When one platform detects a COI it then shares the COI with all of the platforms with which it has a data link. The end of the "Detect" phase for a given target occurs when at least one friendly platform has detected it as a COI.

Our "Track" phase combines the "Fix" and "Track" phases of the generic kill chain model. We do this because tracking a target continuously already includes taking an initial fix. Tracking means continuously maintaining location information on a target with enough precision to engage the target. The track phase is operationally complete when a friendly platform passes the target's coordinates to another friendly platform with a weapon. However, before commencing the engagement of a target, a friendly platform must first successfully identify the target and gain approval for its engagement from higher authority.

"Identification" is how friendly platforms correlate the COI to the desired target. Identification begins after target tracking has commenced, and must conclude before the granting of attack approval.

"Approve" refers to the necessity for permission from higher authority to release ordnance. Approval uses a very complex communication network. For the purposes of this thesis, we assume higher authority grants approval automatically and instantaneously; however, this is a crude approximation of reality.

Once friendly platforms detect the target, pinpoint its location, identify it as the desired target, and receive authorization to release a weapon, a platform may then launch its air-to-ground ordnance. We refer to this as the "Launch" phase.

In the "Control" phase, after a platform launches a weapon, the weapon may require further in-flight target information updates. Some weapons do not require support, but may have functionality for in-flight cueing. The control phase concludes with weapon impact.

In order to determine the success of an attack, friendly platforms must perform a battle damage assessment. We refer to this as the "Assess" phase. Completion of this event concludes the kill chain for this target.

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODEL DESCRIPTION

A. MODEL OVERVIEW

We design the Kill Chain Assessment Tool (KCAT) for use as a scenario-based CONOPS analysis tool with a Microsoft Excel (Microsoft, 2009) interface and Microsoft Excel output.

The interface allows the planner to specify an air-ground strike scenario. The scenario includes choices of friendly platforms, weapons, enemy targets, and their relative positions on a map. The output identifies which platforms and weapons are best to use in the attack scenario, and provides schedules for each kill chain to minimize the total time required to complete the attack mission.

B. AN INTEGER LINEAR PROGRAM TO OPTIMIZE AIR-TO-GROUND STRIKE KILL CHAINS

Because a kill chain involves a sequence of tasks that platforms must accomplish in order, the problem of scheduling platforms to perform the tasks is very much like a manufacturing-style scheduling problem. We draw an analogy between a production scheduling problem and an air-to-ground strike kill chain as follows: we consider a manufacturing "job" to be the prosecution of one target; each available platform is considered a "machine" in the factory; and each phase of the kill chain is considered a "stage" in the production process (Pinedo, 1995).

There are many different types of project scheduling models, but two classes of standard project scheduling problems closely mirror the implementation of a kill chain: the flexible flow shop model and the open shop model.

In a flexible flow shop, there are s stages in series with m machines in parallel at each stage. The flow shop processes jobs first at stage 1, then at stage 2, and so on. A stage functions as a bank of parallel machines; at each stage, job j requires only one machine and, usually, any of the machines in that stage can process job j. This is similar to the air-to-ground strike kill chain in that the phases of the kill chain correlate to the stages of a production process, and there are a number of platforms available at each stage to perform a phase of the kill chain for that target. However, in the flexible flow shop model, the machines in each stage are different from the machines in the other stages. In the air-to-ground strike kill chain, because the machines are platforms that move, and which are re-used throughout all stages of the kill chain, the flexible flow shop model is not an exact match.

The open shop differs from the flexible flow shop in that the routes of the jobs are unspecified, whereas in the flexible flow shop all jobs must follow the same route (Pinedo, 1995). This more closely matches the air-to-ground strike kill chain because now the phases of the kill chain can be assigned to any machine in any order. An elaborate set of constraints then captures the other complex characteristics of the kill chain related to platform and weapon capabilities or limitations.

1. Model Assumptions

A number of model assumptions play a role in our formulation and heuristic solver:

- "No-wait": Transitions from phase to phase in a kill chain for a target proceed without gaps. There is no gap between the "Track" phase and "Identify" phase because then the entire kill chain would have to restart from the "Detect" phase;
- The air-ground target identification rate is 100% when platforms and their associated sensors are within maximum detection range;
- The model uses probability of damage (Pd) values, vice other measures of weapon effectiveness (e.g., "mobility kill" data from a joint munitions effectiveness manual);
- Platforms with sensors capable of providing positive confirmation of target status, such as infrared (IR), electro-optical (EO), or visual image, are able to do so with 100% accuracy when the platform is within detection range; and
- Every launch of a weapon from within the weapon's acceptable launch range results in a successful attack.

We now provide a mathematical formulation of the kill chain optimization problem, KCOPT.

2. Index and Set Use

$p \in P$	platform (alias p')
$t \in T$	target (alias t')
$w \in W$	weapon
$f \in F$	phase of a kill chain (alias f')

$F = \{\text{Detect,Track,Identify,Approve,Launch,Control,Assess}\}$

$(f, f') \in \text{Preclude}$	pairs of distinct kill chain phases that cannot be executed simultaneously by the same platform

3. Data [Units]

$capability_{p,f}$	the maximum number of separate targets on which platform *p* can perform phase *f* (represents data link or sensor capacity of platform)
$process_{p,t,f}$	the minimum amount of time that platform *p* requires in order to perform phase *f* of the kill chain on target *t*
$setup_{p,t,f,t',f'}$	the amount of time that a platform requires to move from performing phase f on target *t* to performing phase *f'* on target *t'*
$winv_p$	the number of weapons carried by platform *p*

4. Decision Variables

$START_{p,t,f}$	the start time that platform *p* begins performing kill chain phase *f* for target *t*
$END_{p,t,f}$	the end time that platform *p* begins performing kill chain phase *f* for target *t*

$$X_{p,t,f} = \begin{cases} 1 & \text{if platform } p \text{ executes kill chain phase } f \text{ on target } t \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{p,t,f,t'f'} = \begin{cases} 1 & \text{if } p \text{ executes } f \text{ on } t \text{ strictly before } f' \text{ on } t' \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{p,t,f',t',f'} = \begin{cases} 1 & \text{if } p \text{ executes } f \text{ on } t \text{ simultaneously with } f' \text{ on } t' \\ 0 & \text{otherwise} \end{cases}$$

5. Formulation KCOPT

$$\min_{START,END,X,Y,Z} C_{\max} \qquad (0)$$

s.t.

$$C_{\max} \geq END_{p,t,Assess} \qquad \forall p,t \qquad (1)$$

$$START_{p,t,f+1} \geq \sum_{p'} END_{p',t,f} - M*\left(1-X_{p,t,f+1}\right) \forall p,t,\, f \neq Assess \qquad (2)$$

$$END_{p,t,f} \geq START_{p,t,f} + process_{p,t,f} * X_{p,t,f} \qquad \forall p,t,\, f \qquad (3)$$

$$START_{p,t',f'} \geq END_{p,t,f} + setup_{p,t,f,t',f'} * Y_{p,t,f,t',f'} \qquad (4)$$
$$-M*(Y_{p,t',f',t,f} + Z_{p,t,f,t',f'}) \qquad \forall p,t,\, f,t',\, f'$$

$$Y_{p,t,f,t',f'} + Y_{p,t',f',t,f} = 1 \qquad \forall\, p,t \neq t',(f,f') \in \text{Preclude} \qquad (5)$$

$$Z_{p,t,f,t',f'} + Y_{p,t,f,t',f'} + Y_{p,t',f',t,f} = 1 \qquad \forall\, p,t \neq t',(f,f') \notin \text{Preclude} \qquad (6)$$

$$\sum_{p} X_{p,t,f} = 1 \qquad \forall t,\, f \qquad (7)$$

$$\sum_{t,t',f'} Z_{p,t,f,t',f'} \leq capability_{p,f} \qquad \forall\, p,\, f \qquad (8)$$

$$\sum_{t} X_{p,t,Launch} \leq winv_{p} \qquad \forall p \qquad (9)$$

6. Discussion

The objective (0) represents the completion time of the "Assess" phase on the last target assessed. Each constraint (1) ensures that the completion of each target's "Assess" phase provides a bound on the overall completion time. Each constraint (2) ensures that the platform performing the next phase of the kill chain on a target does not begin until the current phase has been completed by any platform. Each constraint (3) ensures that the duration of a phase of the kill chain for a target does not exceed the minimum processing time of the platform chosen to perform that phase. Each constraints (4) requires that if a platform is chosen to perform a kill chain phase for a different target, that it cannot begin that phase until a sequence-dependent setup time (e.g., flight time between targets; see below) has been added to the end time of the previous kill chain phase end time.

Each constraint (5) specifies the kill chain phases that must occur sequentially. Each constraint (6) regulates which kill chain phases may occur simultaneously, and which phases must occur sequentially. Each constraint (7) requires for a given kill chain phase for a given target, that exactly one platform performs that function (no duplication of effort). Each constraint (8) requires that a platform not perform the same phase of multiple kill chains for multiple targets beyond its capacity for that phase. Each constraint (9) requires that a platform launch no more weapons than it has in its original weapon inventory.

Other objective functions are, of course, possible. For example, we could calculate average completion time, which is equivalent to minimizing the sum of the end times of all “Assess” phases, or we could minimize the number of targets completed after a fixed deadline using a new set of binary variables that indicate whether the end of the “Assess” phase for target *t* occurs after the deadline, etc.

The most complicated pre-processing that needs to be performed for this model is the determination of the parameter $setup_{p,t,f,t',f'}$ for each platform *p*, pair of targets *t* and *t’*, and pair of phases *f* and *f’*. This computation involves the speed of the platform, the maximum range at which that platform can perform each of the respective phases, and the distance between the targets. A conservative estimate of these sequence-dependent setup costs would assume that the platform begins at the maximum range from target *t* for performing phase *f*, and as far away as possible from target *t’;* it then flies directly towards target *t’* and initiates phase *f’* as soon as it gets within the maximum range for performing *f’* on *t’*.

7. New Data

$speed_p$	the tactical speed of platform *p*, i.e., the speed at which it would fly according to current CONOPS
$rmax_{p,f}$	platform *p*’s maximum range for performing phase *f*
$TTdist_{t,t'}$	the distance between targets *t* and *t’*

We then have the following:

$$setup_{p,t,f,t',f'} = \left(rmax_{p,f} + TTdist_{t,t'} - rmax_{p,f'}\right)^{+} / speed_p$$

where the function $(x)^+$ yields the positive part of x, i.e., the maximum of x and zero. Although each of these calculations is straightforward, and can be made using latitude and longitude data for each target, the number of such calculations is tremendous and, for scenarios of even moderate size (e.g., ten platforms, 30 targets, and seven kill chain phases) can yield models that are too big to be solved as a single, monolithic formulation.

In the presence of SAM sites, the setup times become slightly more complicated. Given a list of SAM sites, indexed by s:

8. New Index and Set

$s \in S$ surface-to-air missile (SAM) site,

and we have a position for each SAM site (latitude and longitude), then the calculation of the setup time for a given platform, flying between two targets, must include any extra time taken to avoid any intervening SAM site, if that platform is vulnerable to surface-to-air missiles.

C. A FAST HEURISTIC

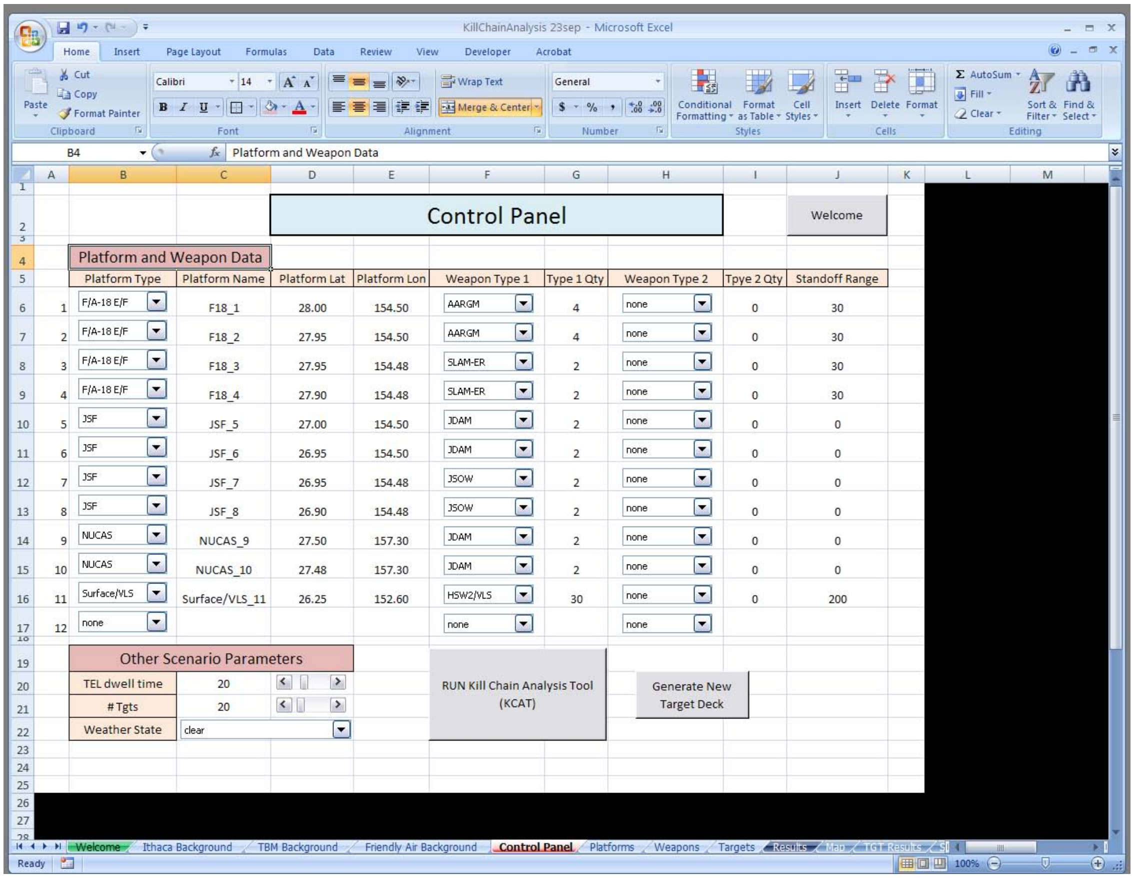
We use a simple constructive heuristic to build feasible schedules for each target.

1. How the Heuristic Works

KCAT is comprised of three subroutines: (1) Getting Data, (2) Building Kill Chains, and (3) Displaying the Results. When KCAT starts, it presents the planner with a welcome screen. From this welcome screen, the planner can choose to peruse several pages of background information related to the Ithaca (or any other) scenario, or the planner can go straight to the "Control Panel."

a. Getting Data

Once at the Control Panel, the planner designs the scenario by selecting platform types, the number of platforms, each platform's starting location, the platforms' weapon configurations, and other data such as standoff ranges, TEL dwell time, weather state, and the number of targets. Figure 2 depicts a screen shot of the Control Panel.



Platform and Weapon Data

	Platform Type	Platform Name	Platform Lat	Platform Lon	Weapon Type 1	Type 1 Qty	Weapon Type 2	Tpye 2 Qty	Standoff Range
1	F/A-18 E/F	F18_1	28.00	154.50	AARGM	4	none	0	30
2	F/A-18 E/F	F18_2	27.95	154.50	AARGM	4	none	0	30
3	F/A-18 E/F	F18_3	27.95	154.48	SLAM-ER	2	none	0	30
4	F/A-18 E/F	F18_4	27.90	154.48	SLAM-ER	2	none	0	30
5	JSF	JSF_5	27.00	154.50	JDAM	2	none	0	0
6	JSF	JSF_6	26.95	154.50	JDAM	2	none	0	0
7	JSF	JSF_7	26.95	154.48	JSOW	2	none	0	0
8	JSF	JSF_8	26.90	154.48	JSOW	2	none	0	0
9	NUCAS	NUCAS_9	27.50	157.30	JDAM	2	none	0	0
10	NUCAS	NUCAS_10	27.48	157.30	JDAM	2	none	0	0
11	Surface/VLS	Surface/VLS_11	26.25	152.60	HSW2/VLS	30	none	0	200
12	none				none		none		

Other Scenario Parameters

TEL dwell time	20
# Tgts	20
Weather State	clear

Figure 2. Screen shot of KCAT Control Panel. In the 'Platform and Weapon Data' section, the planner selects, from left to right, the platform type, the platform's starting latitude and longitude, the platform's first and second weapon type and quantity, and the platform's standoff distance. In the 'Other Scenario Parameters' section, the planner selects TEL dwell time, the number of targets for the scenario, and the scenario weather state. To initiate KCAT, the planner selects "RUN Kill Chain Analysis Tool (KCAT)."

The workbook contains worksheets that include the following: platform data, target data, weapon data, and geographic data.

b. Building Kill Chains

The heuristic is "greedy" in the sense that, at any point in the heuristic when it must decide between platforms to assign to accomplish a specific phase of the kill chain for a given target, it will choose the closest capable platform, where a *capable* platform is one that is able to perform that phase of a kill chain. As the heuristic progresses, it processes phases of the kill chain for each target in sequence, and monotonically in time. Namely, as it assigns platforms to phases of the various kill chains, the heuristic keeps track of the elapsed time in the scenario; as the elapsed time grows, more phases of each kill chain are assigned to platforms and accomplished, and are never re-assigned. Once the heuristic has assigned platforms to all phases of the kill chain for all targets, (and determined start and end times for each phase, for each target), it determines final locations for each platform, and then terminates with a feasible schedule.

KCAT begins by performing an initial feasibility check of the planner-inputted data from the Control Panel. Feasibility pre-processing includes:

- Making sure there are enough weapons for each target;
- Making sure that if a weather state other than clear was selected, then there are enough GPS-guided weapons for each target;
- Making sure that if a platform has a standoff range, then its weapon's maximum range is greater than the standoff range; and
- Making sure that amongst all of the selected platforms, there is enough detection capability to detect each target.

If KCAT determines the scenario is feasible then it loads scenario data into its data structures and calculates the initial distance between each platform and each target and SAM site.

KCAT progresses monotonically through time in the planning horizon. This "stepping though time" is similar to the manner in which discrete event simulation steps in time from one event to the next scheduled event. Time starts at zero; as the algorithm assigns kill chain phases to platforms, the algorithm time-steps forward according to the earliest upcoming event. There are only two types of upcoming events: (1) the completion of a previously assigned task, or, if KCAT did not assign a task, (2) the arrival of a forced time step of 0.1 minutes. If the algorithm assigns multiple tasks, then it steps forward only so far as the earliest completion time out of all the assigned kill chain phases. The minimum time step of 0.1 minutes is our way of handling the case where a platform either starts the scenario out of range of all appropriate targets, or finishes processing one phase of a kill chain, but is not within range of an appropriate target for its next phase. In either case, the heuristic advances that platform towards an appropriate target in 0.1-minute intervals of model time until it is within range of a target. Other mechanisms are certainly possible. Whenever the algorithm takes a step in time, all platforms move towards their closest target a distance equal to the time step multiplied by the platform's speed. Eventually, each platform moves within range of its target and the heuristic assigns it to the next kill chain phase.

However, there are three reasons a platform may not be available for assignment: (1) the platform may lack the capability to perform the phase, (2) it is already handling its maximum number of targets of that phase, or (3) the platform is out of range of the target.

If KCAT successfully finds an available platform, it assigns the desired kill chain phase to that platform. KCAT then determines the next target and phase that correlate to the soonest upcoming event time. In this way, KCAT may skip back and forth in task assignments from (target 6, phase 3) to (target 10, phase 6) to (target 5, phase 4) and so on until all of the assignments have been made.

We gather several values for use in the heuristic from the Control Panel:

$numTargets$	the total number of targets to attack
$total\_p$	the total number of platforms in the strike package
$dwell$	the length of time a TEL target is vulnerable to attack
lat_p	the starting latitude of a given platform
lon_p	the starting longitude of a given platform

We create a number of data structures to maintain parameter values and to assist in our computations. We list a few examples in Table 3.

Name (Parameters)	Interpretation
assignment (t, f)	Whether phase f for target t has been assigned
completion (t, f)	Whether phase f for target t has been assigned
assignmentTime (t, f)	The time at which phase f for target t was assigned
completionTime (t, f)	The time at which phase f for target t was completed
completingPlatform (t, f)	The platform that completed phase f for target t
distance (p, t)	The distance between platform p and target t
SAMdistance (p, s)	The distance between platform p and SAM s
quicknessRatio (p, t)	How quickly platform p's weapon can get to target t
gantt (p)	The collection of tasks assigned to platform p

Table 3. KCAT Data Structures. Each row provides the name of a data structure used in the heuristic and its associated parameters, and its interpretation.

We provide the Building Kill Chains subroutine in pseudo-code below:

```
algorithm build_kill_chains;
begin
    if scenario is infeasible then exit sub
    for each platform p do
        for each target t, calculate distance(p,t) and SAMdistance(p,t)
        calculate quicknessRatio(p,t)
    for each target t do
        assignment(t,0)=TRUE
        assignment(t,8)=FALSE
        completion(t,0)=TRUE
    currentTime, time_stepCounter=0
    all_assigned, all_completed=FALSE
    do while all_completed=FALSE
        something_assigned=FALSE
        for each target t and function f do
```

```
        if currentTime=completionTime(t, f) and
         currentTime<>0 then
                p=completingPlatform(t,f)
                if f<>launch then increment platform function
                 capability
for each platform p do
        calculate closest_target(p); closest_assigned_target(p)
        calculate SAMdistance(p, 0), the distance of the closest
                 SAM s to platform p
for every target t do
        incumbent=0
        if all_assigned=FALSE then
                find the lowest (t, f) pair that needs to be assigned
                if f<=assess then find an incumbent, the closest
                platform p that is capable of performing function f
                for target t
        if incumbent>0 then
                set current task properties
                add current task to gantt (incumbent)
                something_assigned=TRUE
        if incumbent=0 then
                closest_assigned_target(closest platform)=t
for each target t and function f, determine all_assigned,
        all_completed
if all_assigned=FALSE and all_completed=FALSE then
        if something_assigned=TRUE then
                nextTime = the earliest completion time of
                        the tasks assigned on this iteration
        if nextTime-currentTime>1 then
                nextTime=currentTime+1
        if something_assigned=FALSE then
                nextTime=currentTime+0.1
```

```
                if all_assigned=TRUE and all_completed=FALSE then
                    nextTime=the earliest completion time of the tasks that are
                        assigned but not completed in Assess phase
                if all_completed=TRUE then
                    for every platform p and target t do
                        calculate distance from last position
                time_step=nextTime-currentTime
                if time_step>0 then
                    for every platform p do
                        update platform position
                        for every target t do
                            calculate distance(p, t)
                            calculate quicknessRatio(p, t)
                        for every SAM s do
                            calculate SAMdistance(p, s)
                    time_stepCounter=time_stepCounter+1
                if time_stepCounter>1000 then
                    scenario is infeasible; exit sub
            loop
        end
```

c. Displaying the Results

KCAT generates a schedule for each of the platforms and displays all of these on a single worksheet. The schedule details the particulars of each kill chain phase KCAT assigns to that platform. Figure 3 displays a screen shot of the target Gantt chart.

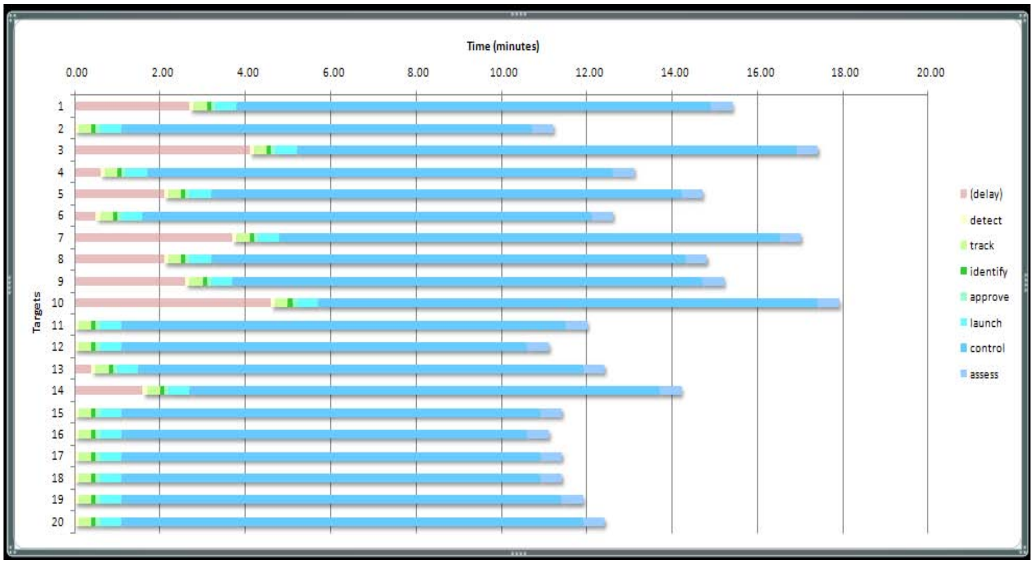


Figure 3. Target Gantt Chart. Each horizontal strip represents the kill chain for the indicated target. Strip colors are color-coded, alternating between dark and light shades, and represent the duration in minutes of an initial idle time, and then the consecutive corresponding kill chain phase. For targets 11, 12, and 15-20, the initial idle time is zero, and so is not shown in this diagram.

2. Feasible Solutions

A feasible solution is an attack schedule that successfully accomplishes every phase of the kill chain for every target. KCAT pre-processes the scenario inputs to rule out infeasible platform, weapon, and target combinations. Additionally, when KCAT processes an attack scenario and advances the elapsed time counter more than 1,000 times (which takes the elapsed time in the model past 100 minutes), the heuristic concludes that a feasible solution is highly unlikely, reports infeasibility, and terminates.

Although KCAT produces feasible schedules, it does not guarantee optimal schedules, or even *successful s*chedules, where we consider a schedule successful if it finishes all kill chains before a given deadline. Recall that, for our example, this deadline is given by the TEL dwell time, and is approximately 20 minutes. The planner can select any TEL dwell time he chooses in the control panel; this value represents the window of vulnerability for all TEL targets in the scenario. If a scenario is feasible, but not successful, KCAT displays the number of kill chains that exceed the TEL dwell time on the “Results” worksheet.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMPUTATIONAL RESULTS

We perform all computations on a 3.16 GHz Dell desktop computer at the Naval Postgraduate School. KCAT run times have never exceeded 2 seconds.

A. BASELINE SCENARIO

KCAT can assess many different models. We illustrate KCAT using a single baseline scenario that looks at an attack in which a package of strike platforms attacks multiple transformer-erector launchers (TELs) within a minimum amount of time.

1. Baseline Scenario Description

The Baseline Scenario consists of 11 friendly platforms and 20 targets. Ten of the friendly platforms are jet aircraft: four F/A-18E/Fs, four Joint Strike Fighters (JSF), and two Navy Unmanned Combat Air Systems (NUCAS). The 11-th friendly platform is a surface vessel with a vertical launch system (VLS). Each of the friendly platforms has a weapons configuration that the planner chooses.

We position the two NUCAS closest to the targets at about 50 nm west of the center of TBM Area 1. Next closest are the division of F/A-18s and the division of JSFs, each at about 200nm west of TBM Area 1. Furthest away is the surface vessel, at about 300 nm from the target area.

Available weapons are Joint Direct Attack Munition (JDAM), Joint Standoff Weapon (JSOW), Advanced Anti-Radiation Guided Missile (AARGM), Standoff Land Attack Missile Extended Range (SLAM-ER), Tomahawk (TLAM), and seven versions of the conceptual High-Speed Weapon (HSW). We examine the utility of HSW, and its impact on the minimization of the longest kill chain duration (i.e., makespan). The platform weapon configuration for the baseline scenario is four AARGM each for the first two F/A-18E/Fs, two SLAM-ER each for the next two F/A-18E/Fs, two JDAM for each of the first two JSF, two JSOW for the latter two JSF, two JDAM for each NUCAS, and 30 HSW version 2 for the surface vessel (Naval Air Warfare Center Weapons Division, 2008).

The twenty targets are mobile TELs. The TELs are located at various positions within a 50-mile radius circle; no pair of TELs is co-located. We assume that the TELs are stationary at the start of the scenario. The TELs are performing a wave attack in which their launch times have been pre-coordinated to occur as close to the same time as possible. We assume all TELs commence launch preparations at the same time, and that there are no differences between TELs with respect to launch preparation times. Therefore, we assume the dwell time for each TEL to be identical.

We choose a TEL dwell time of 30 minutes for KCAT 1.0 evaluation, and 20 minutes for KCAT 1.1 analysis. The standoff range, due to threat integrated air defense systems (IADS), for F/A-18E/Fs is 30nm from any Trinacrian SAMs. JSF and NUCAS do not have associated standoff ranges because we assume they are able to operate without restriction due to their stealth. The weather for the baseline scenario is clear.

2. Blue Forces Concept of Operations (CONOPS)

The forward-positioned NUCAS search for TELs. NUCAS are data-linked to the F/A-18s, JSFs, surface vessel, and to command and control platforms. The NUCAS detect the targets that are in range, and share these locations with all of the other platforms via data link. Each platform either proceeds towards its target or loiters due to standoff distance limitations. As time passes, each platform moves towards its target while executing the phases of the kill chain as required until the scenario is complete.

3. Using KCAT

To run KCAT, the planner specifies scenario information through the control panel (Figure 2). From the control panel, the planner chooses platform types, platform locations, weapon configurations, weapon quantities, platform standoff ranges, TEL dwell times, the number of targets, and the weather state.

The baseline scenario target list is a random draw of 45 targets from 170 possible target launch sites within TBM Area 1. Table 4 lists them:

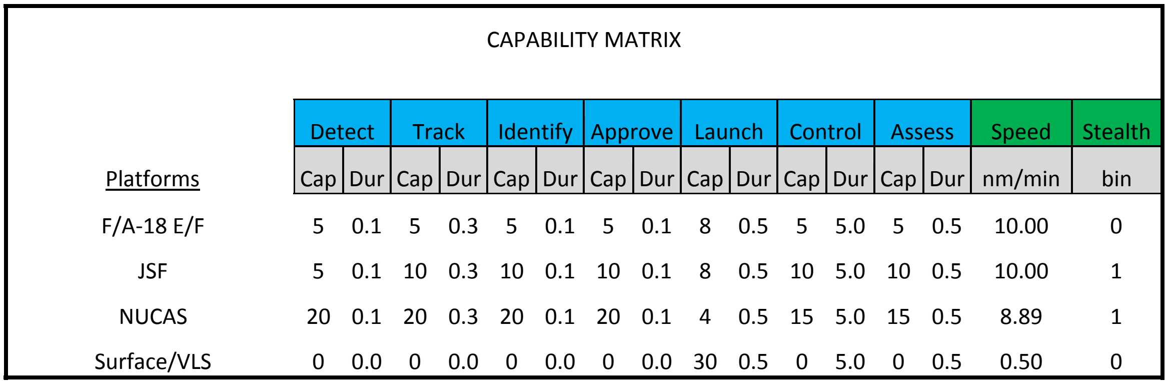
	Target	Lat	Lon	Tgt name	Tgt Description	Tgt Area
1	L120	28.76	158.46	TGT_1	TEL Launch Site	TBM A-1
2	L91	27.73	157.92	TGT_2	TEL Launch Site	TBM A-1
3	L99	28.43	158.99	TGT_3	TEL Launch Site	TBM A-1
4	L50	27.48	158.78	TGT_4	TEL Launch Site	TBM A-1
5	L52	27.22	158.93	TGT_5	TEL Launch Site	TBM A-1
6	L132	28.52	158.16	TGT_6	TEL Launch Site	TBM A-1
7	L3	27.87	159.18	TGT_7	TEL Launch Site	TBM A-1
8	L130	28.44	158.65	TGT_8	TEL Launch Site	TBM A-1
9	L139	28.76	158.40	TGT_9	TEL Launch Site	TBM A-1
10	L121	28.69	158.91	TGT_10	TEL Launch Site	TBM A-1
11	L8	27.48	158.49	TGT_11	TEL Launch Site	TBM A-1
12	L71	27.69	157.90	TGT_12	TEL Launch Site	TBM A-1
13	L147	28.49	158.16	TGT_13	TEL Launch Site	TBM A-1
14	L135	28.34	158.61	TGT_14	TEL Launch Site	TBM A-1
15	L64	27.59	158.12	TGT_15	TEL Launch Site	TBM A-1
16	L164	28.17	157.73	TGT_16	TEL Launch Site	TBM A-1
17	L149	28.37	157.82	TGT_17	TEL Launch Site	TBM A-1
18	L10	27.33	158.18	TGT_18	TEL Launch Site	TBM A-1
19	L162	28.54	158.05	TGT_19	TEL Launch Site	TBM A-1
20	L62	27.89	158.61	TGT_20	TEL Launch Site	TBM A-1
21	L90	27.94	158.07	TGT_21	TEL Launch Site	TBM A-1
22	L131	28.56	158.62	TGT_22	TEL Launch Site	TBM A-1
23	L10	27.33	158.18	TGT_23	TEL Launch Site	TBM A-1
24	L101	28.06	159.22	TGT_24	TEL Launch Site	TBM A-1
25	L80	27.83	158.38	TGT_25	TEL Launch Site	TBM A-1
26	L51	27.87	158.54	TGT_26	TEL Launch Site	TBM A-1
27	L106	28.05	158.90	TGT_27	TEL Launch Site	TBM A-1
28	L111	28.47	158.56	TGT_28	TEL Launch Site	TBM A-1
29	L45	27.45	158.51	TGT_29	TEL Launch Site	TBM A-1
30	L48	27.65	157.98	TGT_30	TEL Launch Site	TBM A-1
31	L142	28.71	158.66	TGT_31	TEL Launch Site	TBM A-1
32	L141	28.10	158.12	TGT_32	TEL Launch Site	TBM A-1
33	L101	28.06	159.22	TGT_33	TEL Launch Site	TBM A-1
34	L168	28.44	157.97	TGT_34	TEL Launch Site	TBM A-1
35	L155	28.50	157.98	TGT_35	TEL Launch Site	TBM A-1
36	L39	27.93	158.69	TGT_36	TEL Launch Site	TBM A-1
37	L119	28.38	158.97	TGT_37	TEL Launch Site	TBM A-1
38	L167	28.33	158.18	TGT_38	TEL Launch Site	TBM A-1
39	L42	27.75	158.60	TGT_39	TEL Launch Site	TBM A-1
40	L91	27.73	157.92	TGT_40	TEL Launch Site	TBM A-1
41	L19	27.72	158.77	TGT_41	TEL Launch Site	TBM A-1
42	L170	28.47	158.52	TGT_42	TEL Launch Site	TBM A-1
43	L115	28.79	158.76	TGT_43	TEL Launch Site	TBM A-1
44	L3	27.87	159.18	TGT_44	TEL Launch Site	TBM A-1
45	L98	27.94	159.19	TGT_45	TEL Launch Site	TBM A-1

Table 4. Baseline Scenario Target List, including locations of each target in area A-1.

We utilize the target list in Table 4 for all baseline scenario analysis. Although KCAT can draw new targets randomly on each run, we do not use this option during testing because the launch area is 50 nm circle, the possible difference in platform-to-target starting distances could be as large as 100 nm.

4. Platform Kill Chain Function Parameters

Different types of platforms have widely varying capabilities. We model these capability differences through the Platform Capability Matrix, which Table 5 depicts.



CAPABILITY MATRIX

	Detect		Track		Identify		Approve		Launch		Control		Assess		Speed	Stealth
Platforms	Cap	Dur	Cap	Dur	Cap	Dur	Cap	Dur	Cap	Dur	Cap	Dur	Cap	Dur	nm/min	bin
F/A-18 E/F	5	0.1	5	0.3	5	0.1	5	0.1	8	0.5	5	5.0	5	0.5	10.00	0
JSF	5	0.1	10	0.3	10	0.1	10	0.1	8	0.5	10	5.0	10	0.5	10.00	1
NUCAS	20	0.1	20	0.3	20	0.1	20	0.1	4	0.5	15	5.0	15	0.5	8.89	1
Surface/VLS	0	0.0	0	0.0	0	0.0	0	0.0	30	0.5	0	5.0	0	0.5	0.50	0

Table 5. Platform Capability Matrix. For the JSF "Detect" capability, the five represents the number of ground targets that a JSF can detect simultaneously, and the 0.1 represents the minimum processing time (minutes) a JSF requires to detect a target. "Track," "Identify," etc. follow in suit. Speed represents the JSF operational speed of 10.0 nm/minute, and the stealth value of 1 represents JSF transparency to threat integrated air defenses (IADS).

The capability matrix lists the four available platforms and their respective capabilities in terms of the seven phases of the kill chain, as well as speed and stealth. The numbers used in this matrix are notional, and unclassified. The planner may amend this matrix as desired to describe platform capabilities more accurately.

Within each phase of the kill chain, each platform has a *capability* and *duration*. Capability refers to the maximum number of targets on which that platform can simultaneously perform that particular function. For example, the surface ship has no capability to detect mobile TELs and therefore its "Detect" capability score is zero, whereas NUCAS has a specialized sensor suite and is rated at 20 simultaneous targets.

Duration refers to the minimum amount of time that a platform needs to perform a function. In “Launch,” for example, all platforms are assigned a duration of 0.5 minutes to conservatively describe a weapon’s launch-to-eject cycle. We assume “Identify” and “Approve” occur instantaneously in this thesis. However, in order to track the progress of these distinct events numerically, we assign them values of 0.1 minutes. We choose 0.1 minutes because this period is very small with respect to typical scenario maximum makespans and thus has very little effect.

The “Control” phase models weapon time-of-flight (TOF). KCAT dynamically computes each platform’s “Control” duration during the execution of the algorithm based on the launch platform’s distance from the target and the speed of the weapon.

We model platform speed as the operational speed: the speed at which the platform would reasonably travel during this mission. Stealth refers to whether the platform is completely transparent to threat IADS.

5. Kill Chain Metrics

We use makespan, or the maximum completion time of all of assigned targets, as our primary metric for analysis. Additionally, KCAT counts the number of targets whose completion time exceeds the TEL dwell time as a measure of overall lateness in the scenario. Finally, KCAT reports the minimum completion time and an average completion time for all of the targets in that scenario.

B. INITIAL RESULTS FOR KCAT VERSION 1.0

Initial results show that KCAT produces feasible schedules. Many of the feasible schedules are clearly not optimal but did indicate trends.

1. KCAT Outputs

After the planner runs KCAT, he sees a results summary screen. From this screen, KCAT presents links to the three outputs: (1) a map display of the scenario, (2) a Gantt chart for the targets, and (3) a schedule for the platforms.

2. Solution Quality

KCAT 1.0 solutions were feasible, but in many cases were clearly not optimal. The makespans for a variety of different scenario configurations were high, on the order of 30–40 minutes. There was little variation in the makespans across a wide spectrum of inputs. Makespans generally decreased when starting parameters were less restrictive (i.e., more platforms were available, or had more weapons, or had weapons with longer ranges, etc.). Since KCAT is a heuristic, we were not surprised to see that, in some of these cases the makespans actually lengthened.

3. Heuristic Problem Areas

Quite a few KCAT 1.0 Gantt charts indicated that schedules were being stacked, i.e., placed end-to-end. This is contrary to the desired function of the heuristic in that a "good" solution would make optimal use of available resources by allocating them in parallel to the maximum extent possible to reduce the scenario makespan. This stacking effect appears to dominate the makespan results for KCAT 1.0.

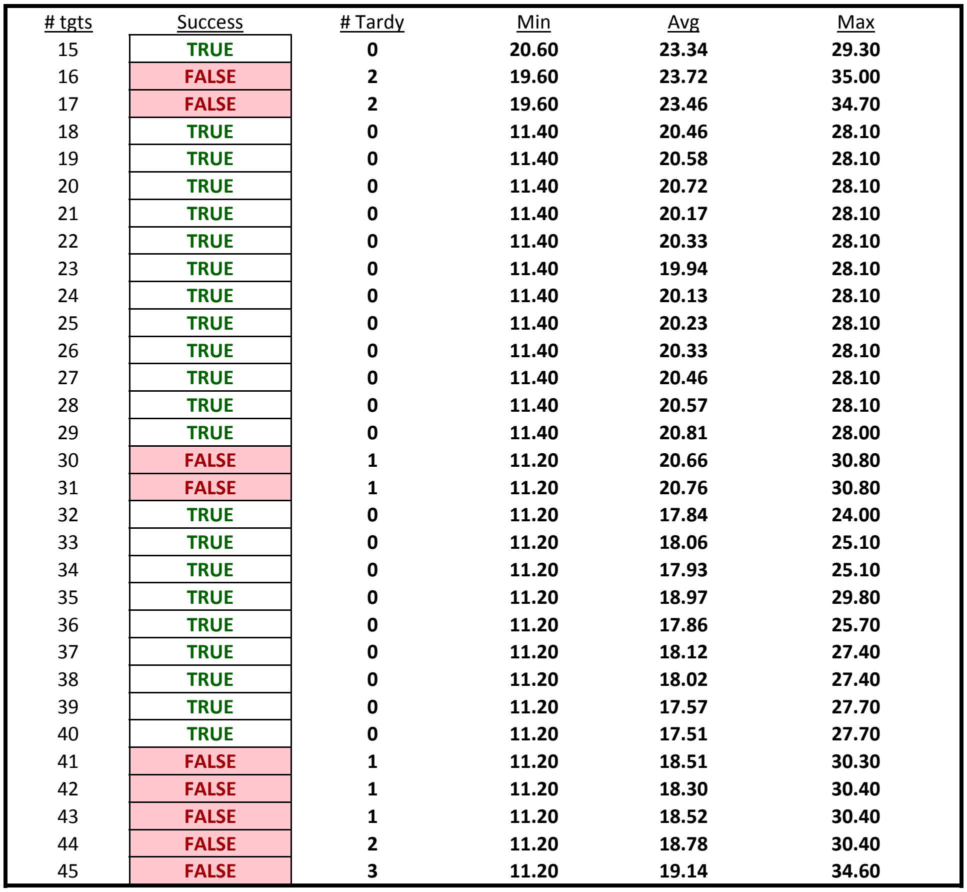
Secondly, Gantt charts indicated that weapon fly-out times were also dominating the makespan, an affect that doubled when stacking occurred. The long weapon flight times even arise in scenarios in which high-speed weapons are available. This indicates that we can improve upon KCAT's method of choosing a platform for the "Launch" phase of the kill chain.

4. Changing the Number of Targets

As the number of targets increases from 15 to 45, makespans generally remain constant at about 28 minutes. The two longest makespans result from scenarios with 16 or 17 targets, which is contrary to expected results. As the number of targets increases from 32 to 45, makespans do exhibit a near-linear increase.

With a 30-minute TEL dwell time, makespans exceed the dwell time at a rate of 9 out of 30 targets. The number of tardy makespans was generally one or two; this appears to be a result of schedule stacking within the algorithm.

Scenario minimum and average times hold very constant at about 11 minutes and 20 minutes, respectively. We highlight the baseline scenario as a consistent frame of reference in yellow. Table 6 summarizes these results.



# tgts	Success	# Tardy	Min	Avg	Max
15	TRUE	0	20.60	23.34	29.30
16	FALSE	2	19.60	23.72	35.00
17	FALSE	2	19.60	23.46	34.70
18	TRUE	0	11.40	20.46	28.10
19	TRUE	0	11.40	20.58	28.10
20	TRUE	0	11.40	20.72	28.10
21	TRUE	0	11.40	20.17	28.10
22	TRUE	0	11.40	20.33	28.10
23	TRUE	0	11.40	19.94	28.10
24	TRUE	0	11.40	20.13	28.10
25	TRUE	0	11.40	20.23	28.10
26	TRUE	0	11.40	20.33	28.10
27	TRUE	0	11.40	20.46	28.10
28	TRUE	0	11.40	20.57	28.10
29	TRUE	0	11.40	20.81	28.00
30	FALSE	1	11.20	20.66	30.80
31	FALSE	1	11.20	20.76	30.80
32	TRUE	0	11.20	17.84	24.00
33	TRUE	0	11.20	18.06	25.10
34	TRUE	0	11.20	17.93	25.10
35	TRUE	0	11.20	18.97	29.80
36	TRUE	0	11.20	17.86	25.70
37	TRUE	0	11.20	18.12	27.40
38	TRUE	0	11.20	18.02	27.40
39	TRUE	0	11.20	17.57	27.70
40	TRUE	0	11.20	17.51	27.70
41	FALSE	1	11.20	18.51	30.30
42	FALSE	1	11.20	18.30	30.40
43	FALSE	1	11.20	18.52	30.40
44	FALSE	2	11.20	18.78	30.40
45	FALSE	3	11.20	19.14	34.60

Table 6. KCAT 1.0 Results, Changing the Number of Targets. From left to right, for 40 targets, the scenario is successful, has 0 tardy kill chains, has a minimum kill chain duration of 11.20 minutes, an average kill chain duration of 17.51 minutes, and a makespan of 27.80 minutes.

5. Changing Platform Weapon Configurations

We examine a number of different weapon configurations, with specific emphasis on looking at the affect of high-speed weapon availability. Model results are counter-intuitive in that replacing slower, shorter-range weapons with faster and longer-range weapons appears to increase or only slightly improve upon the baseline makespan. Table 7 presents these results.

Weapons	Success	# Tardy	Min	Avg	Max
2 HSW1 air / vice F18 SLAM-ER	TRUE	0	9.60	17.88	27.50
2 HSW1 air / vice F18 ARGMM	TRUE	0	9.50	17.60	27.50
2 HSW1 air / vice JSF JDAM	TRUE	0	10.70	19.38	27.30
2 HSW1 air / vice JSF JSOW	FALSE	1	10.80	19.62	31.20
HSW1 VLS	FALSE	11	16.10	28.64	36.70
2 HSW2 air / vice F18 SLAM-ER	TRUE	0	7.00	16.04	23.40
2 HSW2 air / vice F18 ARGMM	TRUE	0	7.00	17.39	25.20
2 HSW2 air / vice JSF JDAM	TRUE	0	7.80	18.21	28.20
2 HSW2 air / vice JSF JSOW	FALSE	1	7.80	18.46	33.10
HSW2 VLS	TRUE	0	11.40	20.72	28.10
2 HSW3 air / vice F18 SLAM-ER	TRUE	0	5.80	16.03	29.70
2 HSW3 air / vice F18 ARGMM	FALSE	1	5.80	18.32	34.40
2 HSW3 air / vice JSF JDAM	TRUE	0	6.40	18.94	29.30
2 HSW3 air / vice JSF JSOW	FALSE	4	6.40	20.18	34.30
HSW3 VLS	TRUE	0	8.80	17.16	21.90
HSW4 VLS	TRUE	0	7.70	18.67	29.50
all JSOW	TRUE	0	5.00	18.97	28.90
all SLAM-ER	FALSE	4	19.50	25.11	33.60
all HSW1 air	FALSE	1	9.50	15.62	31.20

Table 7. KCAT 1.0 Results, Weapon Configuration Variations. In the first row, under ‘Weapons,’ Table 7 indicates that two HSW1 air weapons replace two SLAM-ERs for the F/A-18s in the baseline scenario. Continuing to the right, the scenario is “successful,” the number of tardy kill chains is 0, the duration of the shortest kill chain is 9.60 minutes, the average of all kill chain durations is 17.88 minutes, and the makespan is27.50 minutes.

6. Changing the Availability of a Surface Vessel

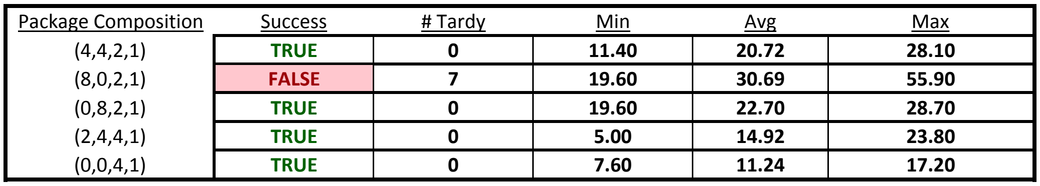
The Baseline Scenario includes one surface vessel with VLS capability. Removing the surface platform from the scenario caused an increase in makespan as expected.

Ship	Success	# Tardy	Min	Avg	Max
no-ship	FALSE	1	19.60	23.34	31.50
ship	TRUE	0	11.40	20.72	28.10

Table 8. KCAT 1.0 Results, Surface Asset Availability. In the first row, for ‘no-ship,’ the scenario is not “successful,” the number of tardy kill chains is one, the duration of the shortest kill chain is 19.60 minutes, the average of all kill chain durations is 23.34 minutes, and the makespan is 31.50 minutes.

7. Changing the Composition of Packages

The baseline scenario contains an attack package of four F/A-18E/Fs, four JSF, two NUCAS, and one surface vessel. We represent this composition in Table 9 as a vector of four numbers, (4, 4, 2, 1), which represents the number of each type of platform, in the order (F/A-18E/F, JSF, NUCAS, surface vessel). We replace the division of JSF by F/A-18s, and as expected (due to a lack of stealth), the makespan increased dramatically and became infeasible with seven tardy kill chains. We then replace the F/A-18s by JSF, and observe slight increases in scenario makespan. Replacing two F/A-18s by two NUCAS results in significant makespan reductions. When the package consists of four NUCAS and one surface vessel, the makespan achieves its lowest average and maximum values. Table 9 describes these results.



Package Composition	Success	# Tardy	Min	Avg	Max
(4,4,2,1)	**TRUE**	**0**	**11.40**	**20.72**	**28.10**
(8,0,2,1)	**FALSE**	**7**	**19.60**	**30.69**	**55.90**
(0,8,2,1)	**TRUE**	**0**	**19.60**	**22.70**	**28.70**
(2,4,4,1)	**TRUE**	**0**	**5.00**	**14.92**	**23.80**
(0,0,4,1)	**TRUE**	**0**	**7.60**	**11.24**	**17.20**

Table 9. KCAT 1.0 Results, Package Composition Variations. In the second row, for the package composition (8, 0, 2, 1), the scenario is not “successful,” the number of tardy kill chains is seven, the duration of the shortest kill chain is 19.60 minutes, the average of all kill chain durations is 30.69 minutes, and the makespan is 55.90 minutes.

8. Changing the Number of Missiles Available to the Surface Vessel

The Baseline Scenario allows 30 missiles for the surface vessel. We decrease this number in an attempt to determine the effects a lesser number of missiles would have on the makespan. KCAT 1.0 results show that the makespan held constant until the number of missiles reached seven, at which point the makespan began to increase significantly. When fewer than four missiles were available, KCAT 1.0 does not reach a feasible solution. Table 10 presents these results.

# of Ship Missiles	Success	# Tardy	Min	Avg	Max
30	TRUE	0	11.40	20.72	28.10
20	TRUE	0	11.40	20.72	28.10
19	TRUE	0	11.40	20.72	28.10
10	TRUE	0	11.40	20.72	28.10
9	TRUE	0	11.40	20.83	28.10
8	TRUE	0	11.40	20.87	28.10
7	FALSE	1	11.40	21.27	30.70
6	FALSE	1	11.40	21.34	30.20
5	FALSE	1	11.40	21.51	30.20
4	FALSE	3	11.40	22.33	37.00
3	FALSE				
2	FALSE				
1	FALSE				

Table 10. KCAT 1.0 Results, Number of Available Missiles to Surface Platform. In the first row, for 30 missiles, the scenario is "successful," there are zero tardy kill chains, the duration of the shortest kill chain is 11.40 minutes, the average of all kill chain durations is 20.72 minutes, and the makespan is 28.10 minutes

9. Heuristic Improvements

The two main areas for improvement in KCAT 1.0 relate to (1) the observed stacking of kill chain schedules, resulting in excessive and counter-intuitive makespan results, and (2) the heuristic's method of choosing a launch platform, which resulted in the choice of slower weapons even in the presence of high-speed weapons.

We address the stacking problem by forcing the heuristic to stop at each one-minute interval to "look" for platforms to perform kill chain phases. In KCAT 1.0, the heuristic moves monotonically forward in time to the next scheduled event. While this works well for kill chain phases of short duration, phases of long duration (i.e., the "Control" phase in which durations could be as large as 18 or 20 minutes), cause a "blank" period. During the blank period, the heuristic will not attempt to assign kill chain phases and thus creates the observed stacking anomaly.

We correct KCAT 1.0's choice of slower weapons amidst higher-speed weapons by revising the heuristic's "greedy" platform assignment process. Instead of basing platform assignments on platform distance to a prospective target, we base it on platform *quickness*, where quickness is the ratio of a platform's distance to a target to platform speed. In other words, if one platform is twice as fast, but is located one mile further from a target than another platform, we desire the heuristic to choose the further platform because it can perform the kill chain phase more quickly. Platform "quickness" is especially important when KCAT chooses a platform for the "Launch" phase of a kill chain. This is because, depending on the weapon, the time of flight can be quite large and may thus dominate the makespan.

Because the baseline scenario assumes that all of the platforms travel at similar speeds, distance works well as a surrogate for quickness for all phases of the kill chain except for the "Launch" phase. For the "Launch" phase, weapon speed is very important to the weapon's total time-of-flight. Therefore, we develop a "quickness ratio," which is a platform's distance to the target divided by the speed of the platform's weapon. In this way, KCAT 1.1 now chooses the platform that is able to get its weapon to the target the quickest for "Launch" phases, whereas in the other phases it chooses the closest platform.

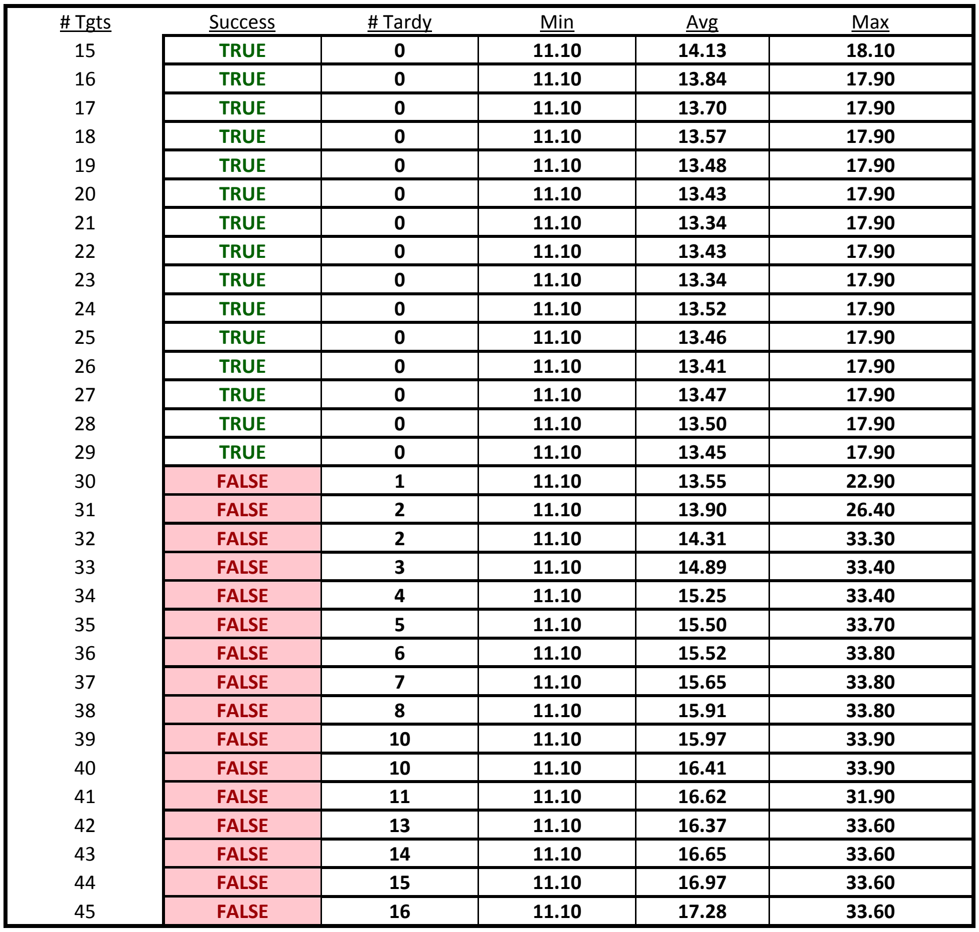
C. KCAT VERSION 1.1 RESULTS

KCAT 1.1 achieves feasible results that are superior to KCAT 1.0. For the baseline scenario, the KCAT 1.1 makespan was 17.9 minutes as compared to 28.1 minutes for KCAT 1.0. This represents a 36.3% improvement. With respect to average completion times across a variation in the number of targets, KCAT 1.1 averages approximately 15 minutes, while KCAT 1.0 averages about 20 minutes. This represents a 25% improvement in average completion time.

Because the KCAT 1.1 consistently produced makespans spanning fewer than 20 minutes, we reduce the TEL dwell time to 20 minutes vice 30 minutes. We also relax the Detect, Track, Identify, Approve, and Assess range constraints from 50 nm to 75 nm to more accurately model NUCAS target interaction.

1. Changing the Number of Targets

As the number of targets increased from 15 to 45, KCAT 1.1 makespan held steady at 17.9 minutes until reaching 30 targets. At 30 targets, we observe a steep increase to makespans of about 34 minutes. We attribute this to the expenditure of all of the surface vessel's high-speed weapons. Table 11 and Figure 4 present this information.



# Tgts	Success	# Tardy	Min	Avg	Max
15	TRUE	0	11.10	14.13	18.10
16	TRUE	0	11.10	13.84	17.90
17	TRUE	0	11.10	13.70	17.90
18	TRUE	0	11.10	13.57	17.90
19	TRUE	0	11.10	13.48	17.90
20	TRUE	0	11.10	13.43	17.90
21	TRUE	0	11.10	13.34	17.90
22	TRUE	0	11.10	13.43	17.90
23	TRUE	0	11.10	13.34	17.90
24	TRUE	0	11.10	13.52	17.90
25	TRUE	0	11.10	13.46	17.90
26	TRUE	0	11.10	13.41	17.90
27	TRUE	0	11.10	13.47	17.90
28	TRUE	0	11.10	13.50	17.90
29	TRUE	0	11.10	13.45	17.90
30	FALSE	1	11.10	13.55	22.90
31	FALSE	2	11.10	13.90	26.40
32	FALSE	2	11.10	14.31	33.30
33	FALSE	3	11.10	14.89	33.40
34	FALSE	4	11.10	15.25	33.40
35	FALSE	5	11.10	15.50	33.70
36	FALSE	6	11.10	15.52	33.80
37	FALSE	7	11.10	15.65	33.80
38	FALSE	8	11.10	15.91	33.80
39	FALSE	10	11.10	15.97	33.90
40	FALSE	10	11.10	16.41	33.90
41	FALSE	11	11.10	16.62	31.90
42	FALSE	13	11.10	16.37	33.60
43	FALSE	14	11.10	16.65	33.60
44	FALSE	15	11.10	16.97	33.60
45	FALSE	16	11.10	17.28	33.60

Table 11. KCAT 1.1 Results, Changing the Number of Targets. From left to right, for 39 targets, the scenario is not "successful," there are 10 tardy kill chains, the duration of the shortest kill chain is 11.10 minutes, the average of all kill chain durations is 15.97 minutes, and the makespan is 33.90 minutes.

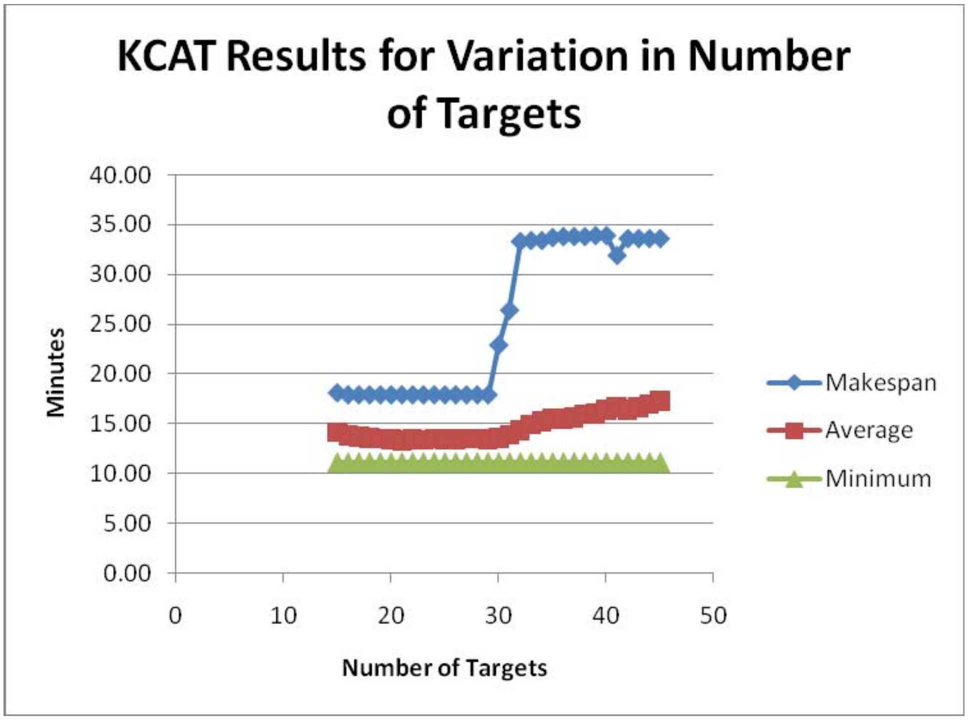


Figure 4. KCAT Results, Variation in the Number of Targets. For 20 targets, the minimum kill chain duration is approximately 10 minutes, the average kill chain duration is approximately 14 minutes, and the makespan is approximately 18 minutes.

2. Changing Platform Weapon Configurations

Changes in the weapon configuration of the F/A-18s and JSF did not affect the makespan at all in most cases. We attribute this result to the geographic positioning of the platforms and targets at the start of the scenario.

When the surface vessel launched High-Speed Weapon version 1 instead of the faster version 2, the makespan increased significantly and no longer finished within the TEL dwell time of 20 minutes. We display KCAT 1.1 results for changes in weapon configurations in Table 12.

Weapon Configuration	Success	# Tardy	Min	Avg	Max
2 HSW1 air / vice F18 SLAM-ER	**TRUE**	**0**	**10.00**	**13.25**	**17.90**
2 HSW1 air / vice F18 ARGMM	**TRUE**	**0**	**9.50**	**12.97**	**17.90**
2 HSW1 air / vice JSF JDAM	**TRUE**	**0**	**10.20**	**13.28**	**17.90**
2 HSW1 air / vice JSF JSOW	**TRUE**	**0**	**10.30**	**13.30**	**17.90**
HSW1 VLS	**FALSE**	**6**	**15.60**	**18.45**	**23.40**
2 HSW2 air / vice F18 SLAM-ER	**TRUE**	**0**	**7.30**	**12.68**	**17.90**
2 HSW2 air / vice F18 ARGMM	**TRUE**	**0**	**7.30**	**12.68**	**17.90**
2 HSW2 air / vice JSF JDAM	**TRUE**	**0**	**7.40**	**12.70**	**17.90**
2 HSW2 air / vice JSF JSOW	**TRUE**	**0**	**7.50**	**12.71**	**17.90**
HSW2 VLS	**TRUE**	**0**	**11.10**	**13.43**	**17.90**
2 HSW3 air / vice F18 SLAM-ER	**TRUE**	**0**	**6.00**	**12.40**	**17.90**
2 HSW3 air / vice F18 ARGMM	**TRUE**	**0**	**6.00**	**12.40**	**17.90**
2 HSW3 air / vice JSF JDAM	**TRUE**	**0**	**6.10**	**12.42**	**17.90**
2 HSW3 air / vice JSF JSOW	**TRUE**	**0**	**6.20**	**12.43**	**17.90**
HSW3 VLS	**TRUE**	**0**	**8.60**	**10.64**	**14.80**
HSW4 VLS	**TRUE**	**0**	**7.50**	**9.44**	**13.50**
all JSOW	**TRUE**	**0**	**11.10**	**13.43**	**17.90**
all SLAM-ER	**TRUE**	**0**	**11.10**	**13.43**	**17.90**
all HSW1 air	**TRUE**	**0**	**9.60**	**12.67**	**17.90**

Table 12. KCAT 1.1 Results, Changing Platform Weapon Configurations. In the first row, under ‘Weapons,’ Table 12 indicates that two HSW1 air weapons replace two SLAM-ERs for the F/A-18s in the baseline scenario. Continuing to the right, the scenario is “successful,” the number of tardy kill chains is zero, the duration of the shortest kill chain is 10.00 minutes, the average of all kill chain durations is 13.25 minutes, and the makespan is 17.90 minutes.

Figures 5 and 6 are the Gantt chart and scenario map, respectively, for the scenario in which all F/A-18s and JSF are loaded with High-Speed Weapon version 1. From Table 12, we see that the makespan for this scenario was 17.9 minutes. This does not improve upon the baseline scenario (a counter-intuitive result).

Examination of the scenario map and Gantt chart for this scenario reveals a correlation between target distance and makespan. Targets 1, 3, 7, and 10, the four targets with the longest completion times in the scenario, are also four of the six easternmost targets geographically.

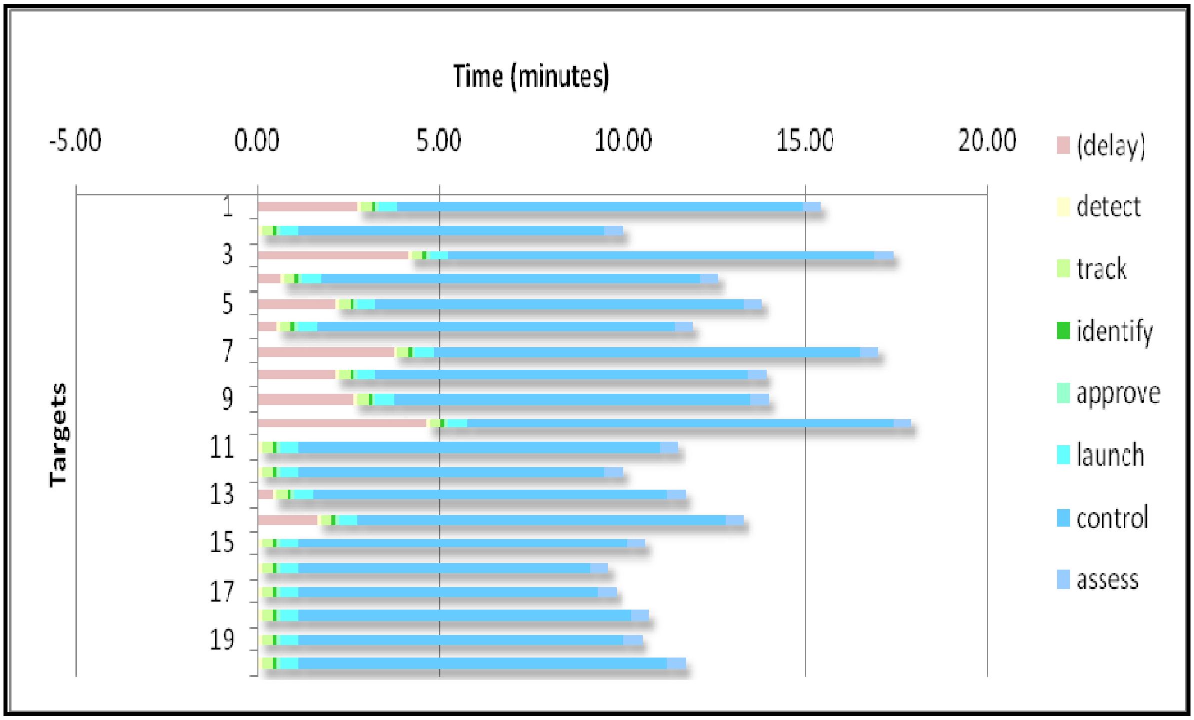


Figure 5. KCAT 1.1 Gantt Chart. For target 1, the Detect phase begins at approximately 3 minutes after the start of the scenario. The Track, Identify, Approve, and Launch phases take short amounts of time and complete after approximately 4 minutes of total elapsed time. The control phase, representing weapon TOF extends from 4 minutes until 15 minutes. Assess begins at 15 minutes and concludes at approximately 16 minutes.

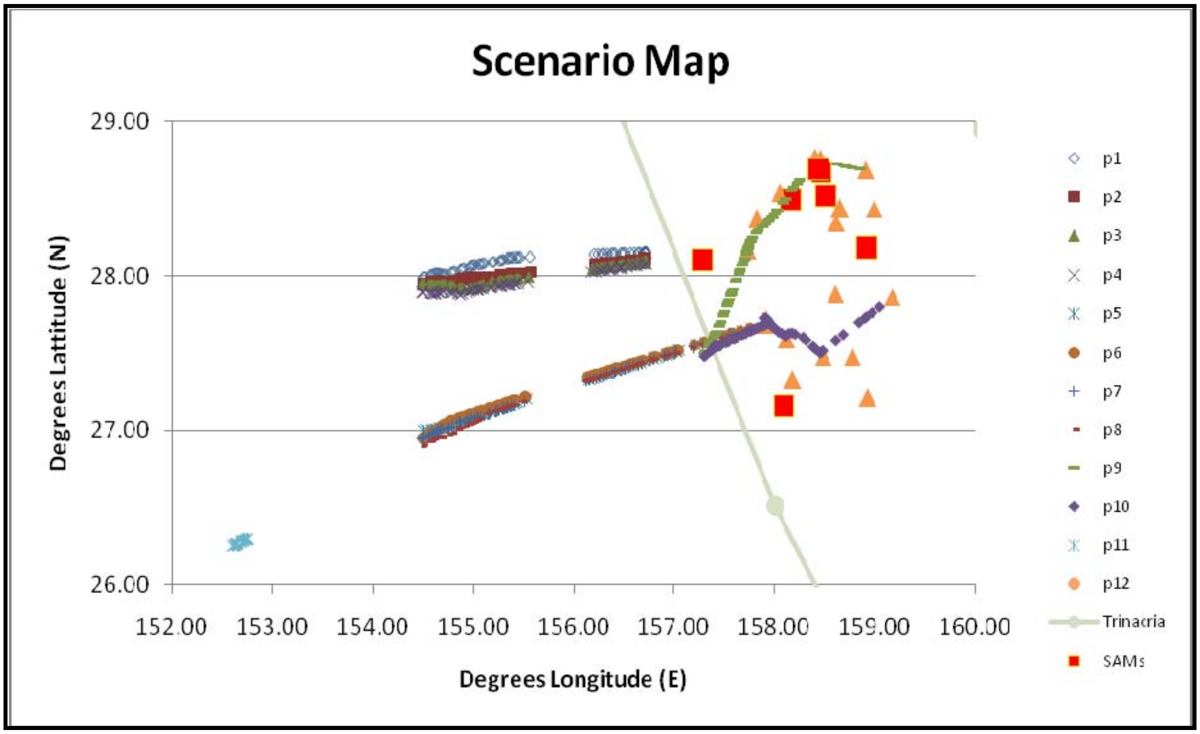


Figure 6. KCAT 1.1 Scenario Map. The scenario map displays the positions in degrees latitude and longitude of the platforms, targets, and SAMs in the scenario. The map displays position data for times at which a kill chain phase is beginning or ending for any of the targets.

3. Changing the Availability of a Surface Vessel

We remove the surface vessel with VLS from the scenario, and observe that the makespan increases as expected. The scenario no longer accomplishes the kill chain during the TEL dwell time. Table 13 presents these results.

Ship	Success	# Tardy	Min	Avg	Max
no-ship	**FALSE**	**2**	**9.60**	**13.37**	**24.20**
ship	**TRUE**	**0**	**11.10**	**13.43**	**17.90**

Table 13. KCAT 1.1 Results, Availability of a Surface Vessel. For 'no-ship,' the scenario is not "successful," the number of tardy kill chains is two, the duration of the shortest kill chain is 9.60 minutes, the average of all kill chain durations is 13.37 minutes, and the makespan is 24.20 minutes.

In Figure 7, we observe that for this scenario KCAT assigns a platform to perform "Track" for a long time before assigning weapon launch. KCAT vectors the two NUCAS, which are located furthest east at the start of the scenario, to hit the furthest targets with their JDAM.

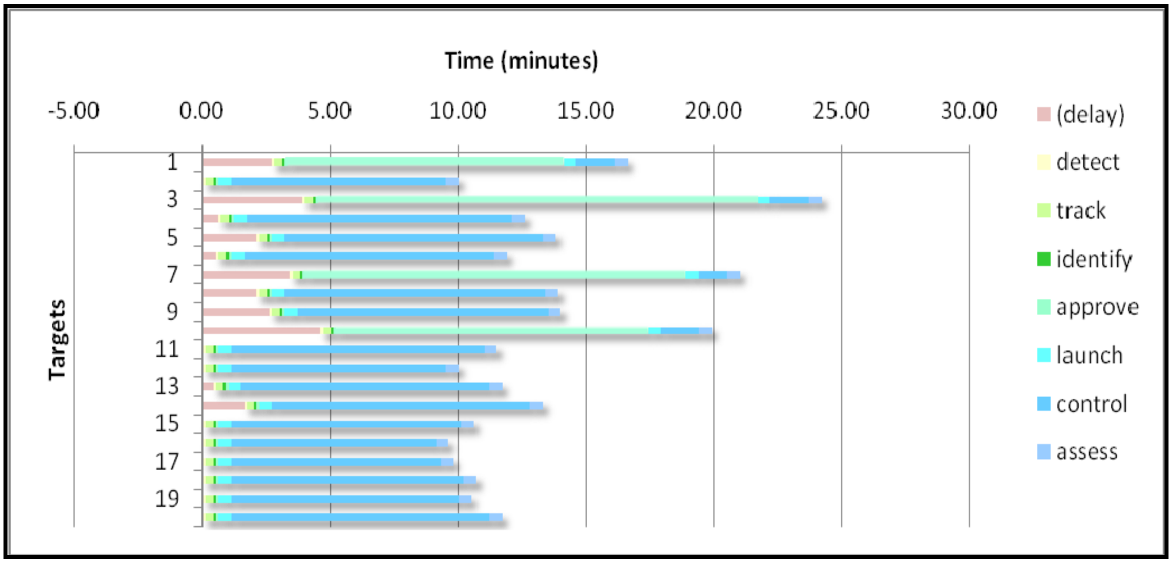


Figure 7. KCAT 1.1 Gantt Chart Depicting Scenario Without a Surface Vessel.

4. Changing the Composition of Packages

When the surface vessel is present and armed with High-speed weapons, KCAT chooses it as the launching vessel in almost all cases. When NUCAS is present, KCAT chooses it to perform almost all of the other kill chain functions. For this reason, removal of all of the F/A-18E/Fs and JSF does not affect the makespan of the scenario significantly. Several NUCAS and one surface vessel appear to be nearly as effective as the full package of platforms in the baseline scenario. Table 14 shows these results.

package composition	Success	# Tardy	Min	Avg	Max
(4,4,2,1)	TRUE	0	11.10	13.43	17.90
(8,0,2,1)	TRUE	0	11.10	13.43	17.90
(0,8,2,1)	TRUE	0	11.10	13.43	17.90
(2,4,4,1)	TRUE	0	11.10	13.43	17.90
(0,0,4,1)	TRUE	0	11.10	13.40	18.30

Table 14. KCAT 1.1 Results, Changing the Composition of Platforms. In the first row, the package composition is (4, 4, 2, 1). The scenario is "successful," the number of tardy kill chains is zero, the duration of the shortest kill chain is 11.10 minutes, the average of all kill chain durations is 13.43 minutes, and the makespan is 17.90 minutes

5. Changing the Number of Missiles Available to the Surface Vessel

As we decrease the number of missiles available to the surface vessel, the makespan increased. Because the surface vessel is the preferred "Launch" platform, and because there are 20 targets in the baseline scenario, the makespan increases significantly when the surface vessel had fewer than 20 missiles available for "Launch." As the number of missiles decreased further, the makespan continued to increase a slower rate. Figure 8 displays these results.

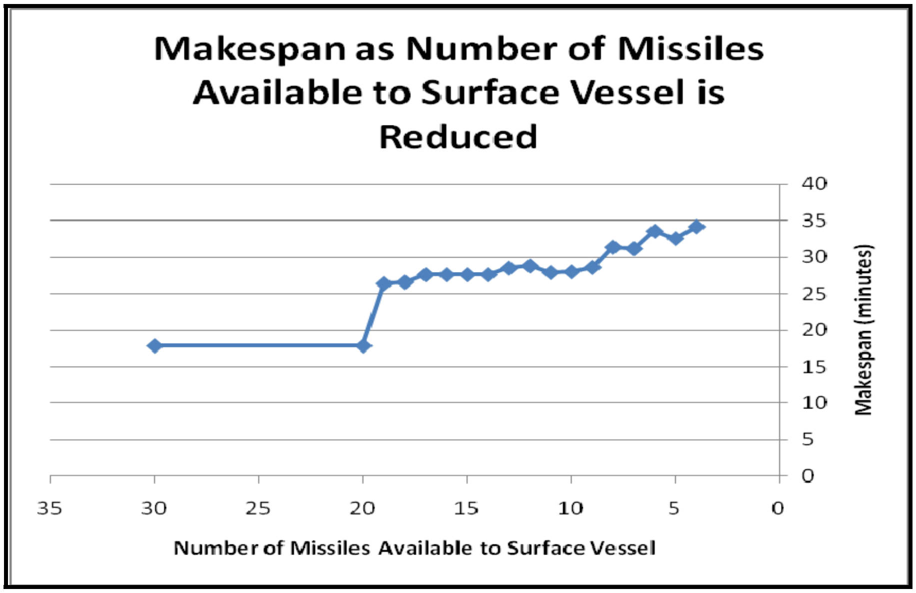


Figure 8. KCAT 1.1 Results, Changing the Number of Missiles Available to the Surface Vessel. When 20 missiles are available, the makespan is approximately 18 minutes.

6. Changing the Target Deck for the Baseline Scenario

Makespans vary little in the baseline scenario as the target deck changes. We perform 40 random samples from the pool of 170 TEL launch site locations within TBM Area 1 to investigate how the geographic arrangement of targets affects the makespan. Each random sample consists of 20 targets (the number of targets in the baseline scenario). The mean makespan of the data is 21.27 minutes +/- 0.60 minutes. We observe a maximum makespan of 27.7 minutes, and a minimum makespan of 16.6 minutes. The baseline scenario's makespan of 17.9 minutes is within one standard deviation of the computed sample mean.

7. Heuristic Problem Areas and Possible Improvements

Although KCAT 1.1 produces makespans of much shorter duration than KCAT 1.0, one significant area for improvement remains. The logic in KCAT that assigns the direction that a platform moves is simplistic and can lead to extended makespans, or possibly infeasibility due to the number of iterations. When KCAT assigns a platform to one or more kill chain phases, it moves towards the closest target for which it is performing the kill chain phase. When a platform is not assigned a kill chain phase, it moves towards whichever target is closest geographically.

In some cases, KCAT assigns platforms to a target that is located far from the other targets, and in this manner, steers the platform away from areas where KCAT could utilize the platform's capabilities more effectively. If KCAT's platform capability settings are set relatively low, at 30 nm for example, then as the platform heads away from a group of targets it may become ineligible to participate in kill chains for those targets. Improving KCAT's platform directive logic could thus lead to significantly improved results.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

In this thesis, we have described KCAT (Kill Chain Assessment Tool), an assessment tool for air-to-ground kill chain analysis. KCAT comprises a Microsoft Excel-based graphical user interface (GUI), a heuristic solver, and three output products.

B. OPERATIONAL APPLICATIONS

KCAT enables the Warfare Analysis and Integration Branch of Naval Air Systems Command (NAVAIR), Naval Air Weapons Station (NAWS) China Lake, CA, to perform a multitude of platform and weapon analyses. It addresses one of the most difficult challenges in warfare today: time-critical strike.

Active defense strategies garner most of today’s activity in the realm of theater ballistic missile defense (TBMD). KCAT offers insights into the just-as-important strategy of counterforce. KCAT results can inform the larger picture of TBMD, in which both active defense and counterforce strategies are important components.

KCAT is capable of handling current and contemplated future weapon platforms and weapons. KCAT allows the planner to make performance comparisons between similar versions of the same weapon system, or between vastly different systems. The planner may use these comparisons as a foundation for analysis of alternatives studies.

Although set near the fictitious island Ithaca, the planner may adjust KCAT quite easily to represent any MCO in existence today. Additionally, the planner can alter weapon and platform parameters to more accurately represent actual platform and weapon functionality at other classification levels.

KCAT produces feasible results in two seconds or less. Results are not optimal, but do provide insights and indicate trends. KCAT runs on any computer system with Microsoft Excel, and is scalable to handle scenarios anywhere from a few platforms and targets to 12 platforms and 45 targets.

C. FUTURE DEVELOPMENT

KCAT will ideally generate provably optimal solutions. Future developments fall into two categories: heuristic algorithm improvements, and broadening KCAT functional capabilities.

1. Heuristic Improvements

We could improve KCAT’s solutions by using them as starting solutions in a neighborhood search algorithm that makes intelligent schedule swaps, such as exchanging two platforms between targets to reduce their overall travel times, or swapping two phases of different targets processed by the same platform to reduce that single platform’s flight time. These schedule swaps are “good” if they improve the solution in terms of further minimizing the scenario makespan.

Eventually, we wish to build upon the integer linear programming formulation in Chapter III and successfully develop an integer linear program that produces provably optimal solutions.

Future work could reduce scenario makespans by implementing a more effective mechanism to govern platform routing commands. One option is to route platforms based on the geographic arrangement of all of the targets instead of on just the closest target.

2. Functional Improvements

In its current state, KCAT includes four Navy or joint platforms. A broadening in the choice of platforms to include Air Force and possibly Marine Corps and Army units would be beneficial considering the joint nature of current and future operations.

Similarly, we could increase the variety of weapons available to planners, such as those used by the Air Force, to broaden the types of scenarios for which KCAT is appropriate. We did not consider the effect of GPS jamming in this thesis. Future work towards quantification of these effects would definitely be worthwhile. Similarly, future work can take into account weather constraints.

We did not model network connectivity in this thesis. Future work could entail implementation of network connectivity constraints in both the integer linear program and in the heuristic solver to ensure that KCAT maintains network connectivity for each platform that requires it.

Future work can model the details of the “Identification” and “Approval” phases of the kill chain. For the Baseline Scenario, in which the targets are already well defined, identification and approval are straightforward. However, in other forms of time-critical targeting, identification and approval can be the most difficult portions of the kill chain.

Finally, we believe future work can include improving the graphical user interface (GUI) of KCAT both in terms of inputting scenario data and in terms of improving output products. Future work can apply existing commercial software to improve upon KCAT Gantt chart, scenario map, and platform schedule output displays.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Brickner, W. K. (2005). An Analysis of the Kill Chain for Time-Critical Strike. MS Thesis in Operations Research. Monterey, CA: Naval Postgraduate School.

Brown, G., Carlyle, M., Diehl, D., Kline, J., and Wood, K. (2005). A Two-Sided Optimizaiton for Theater Ballistic Missile Defense. *Operations Research , 53* (5), 745–763.

Defense Science Board. (2009). *Report of the Defense Science Board Task Force on Time Critical Conventional Strike from Strategic Standoff.* Washington, D.C.: Defense Science Board.

Feickert, A. (2005). *Missile Survey: Ballistic and Cruise Missiles of Selected Foreign Countries.* Washington, D.C.: Congressional Research Service.

IBM. (2009, September 28). *IBM Rational System Architect*. Retrieved from http://www.telelogic.com/products/systemarchitect/index.cfm

Loibl, D. (2008). Ithaca: An Unclassified Scenario Suitable for International Use. China Lake, CA: Warfare Analysis & Integration Department, Naval Air Warfare Center.

Marshall. (1994). Quantifying Counterforce and Active Defense in Countering Theater Ballistic Missiles. *Military Operations Research* (Winter), 34–48.

Microsoft. (2009, September 25). *Microsoft Excel*. Retrieved from http://office.microsoft.com/en-us/excel/FX100646951033.aspx

National Air and Space Intelligence Center. (2009). *Ballistic and Cruise Missile Threat.* Wright-Patterson AFB.

Naval Air Systems Command, PMA-201. (2006). PMA-201 Kill Chain Analysis Process. 1–16.

Naval Air Warfare Center Weapons Division. (2008). Naval Weapons Handbook. *(Revision 4) , NAWCWD Technical Publicaiton 8451*. China Lake, CA.

Pinedo, M. (1995). *Scheduling: Theory, Algorithms, and Systems.* Englewood Cliffs, NJ: Prentice Hall.

Smith, R. D. (2008). Kill Chain Scoring Criteria. 1–23. China Lake, CA: Naval Air Systems Command.

Wilson, G. (2002). A Time-Critical Targeting Roadmap. Maxwell Air Force Base, AL: Air Command and Staff College, Air University.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Distinguished Professor Gerald Brown
   Naval Postgraduate School
   Monterey, CA

4. Professor W. Matthew Carlyle
   Naval Postgraduate School
   Monterey, CA

5. William K. Brickner
   Head, Platform Weaponization Branch, AIR 4.10.2.3
   Warfare Analysis & Integration Department
   China Lake, CA

6. Ken Amster
   Senior Analyst, Warfare Analysis Division
   Naval Air Warfare Center, Weapons Division
   China Lake, CA