A Discrete Event Simulation of Network Centric Operations: Modeling Unbalanced Combat Configurations in Symmetric Engagements

Mahmoud Turki Khasawneh
Old Dominion University

A DISCRETE EVENT SIMULATION OF NETWORK CENTRIC OPERATIONS: MODELING UNBALANCED COMBAT CONFIGURATIONS IN SYMMETRIC ENGAGEMENTS

by

Mahmoud Turki Khasawneh

B.S. December 2007, The Hashemite University, Jordan
M.E. May 2009, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ENGINEERING MANAGEMENT

OLD DOMINION UNIVERSITY
May 2012

Approved by:

Ghaith Rabadi (Director)

Shannon R. Bowling (Member)

Resit Unal (Member)

Sean Deller (Member)

ABSTRACT

A DISCRETE EVENT SIMULATION OF NETWORK CENTRIC OPERATIONS: MODELING UNBALANCED COMBAT CONFIGURATIONS IN SYMMETRIC ENGAGEMENTS

Mahmoud Turki Khasawneh
Old Dominion University, 2012
Director: Dr. Ghaith Rabadi

Network Centric Operations (NCO) has been dubbed the most significant revolution in military affairs (RMA) in the past 200 years. The promise of NCO is based on the notion that information sharing and connectivity is fundamental to the effectiveness of a combat force. This due to the ability of a properly networked force to self-synchronize itself as it engages enemy forces. The purposeful arrangement of assets in a combat force is what makes it 'properly networked'. What is a purposeful arrangement of combat assets? How should a force organize to enhance information sharing and connectivity? And how does connectivity within a networked force impact its combat effectiveness? This research builds a discrete-event simulation of the information age combat model, which is a representation of NCO, in an attempt to understand the impact of information sharing and connectivity among the elements of a military force on its combat effectiveness. Unbalanced combat configurations doing symmetric engagements were selected as the prime focus. They were studied and simulated to gain insights into the dynamics of networked operations. The proposed discrete event combat model displayed significant increases in efficiency and speed of running compared to previous modeling work that utilized agent-directed simulations. Linear and nonlinear regression analyses were conducted to highlight the performance metrics that wield significant predictive power over the probability of winning a combat engagement.

I would like to dedicate my dissertation to Allah almighty. My faith in him is the guide that has been showing me the path all my life. Everything I am, and everything I wish to be, is because of Allah, and I thank him for the blessings he bestowed upon me, and all those that await me. Thank you, Allah. Without you, I am a lost soul.

I would also like to dedicate my dissertation to my family. I would have never completed this phase of my life if it was not for their endless support, sacrifice, and prayers. To my Dad, Major General (Ret.) Turki M. Khasawneh, and my Mom, En'am Al-Wahshat, I am at a loss for words to describe my gratitude towards you both. Should I be able in the future to teach my children a small portion of the values you instilled in me, I will be very proud of myself. And should I be able to have my children love me the way me and my brothers and sister love you, then I will be a very proud parent. You have always been the light at the end of the tunnel, you have never let me down, and you have always gave me your hand to hold to and stand up every time I tripped on one of those bumps in the long highway we call life. Nothing makes me happy like seeing that smile on your faces. Your love is the only wealth I truly own in this world. Thank you for everything. I love you with all my heart. To my brothers, Dr. Mohammad Khasawneh, Dr. Fadi Khasawneh, Shadi Khasawneh, Omar Khasawneh, and my sister, Rasha Khasawneh, you are the ones who carried me through good times and hard times. You are the shoulder I lean on. Whenever I feel burdened, you never hesitate to carry that burden with me. I cannot ask for a better circle of support, because you are everything I wish for, and even more. I owe you so much, although you always tell me that is not true. Thank you so much for everything.

ACKNOWLEDGMENTS

There are so many people who have contributed to the successful completion of this dissertation. I extend many, many thanks to my advisers Dr. Shannon Bowling and Dr. Ghaith Rabadi. This work would have never been completed without your guidance and support. I have known you for five years, but it feels like I have known you for an eternity. I am indebted to you for the rest of my life. I would also like to thank my committee members, Dr. Resit Unal and Dr. Sean Deller, for their valuable contribution and advice. The final product was greatly improved because of your contribution and I am very grateful that you served on my committee. I want to also take this opportunity to acknowledge my dear friend and colleague, Nevan Shearer, who collaborated with me on the verification of the discrete event combat model, which served as a starting point for each one of us to pursue an independent research study related to the information age combat model. I would also like to thank my friends in the United States and back home in Jordan who made the journey towards my doctorate a much more tolerable experience. Amr Alnsour, Ahmed Hamdy, Jun Zhang, Nevan Shearer, Raed Jaradat, Sawsan Abutabanjeh, Mohammad Obeid, Mariam Kotachi, Tariq Tashtoush, Yaw Mensah, Berna Eren Tokgoz, Max Siangchokyoo, and many more friends, thank you very much for your friendship and support all those years. I know I am doing well in my life since I am surrounded by wonderful people like you all.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

There is no doubt that information technology has revolutionized all aspects of our life. Its applications are everywhere to be seen, whether it is in economics, sociology, medicine, telecommunications, politics, warfare, travel, or the everyday life of human beings. The numerous applications of information technology, though in different disciplines or areas, have one far-reaching consequence in common; the proliferation of knowledge and information is at unprecedented levels. The world has never been 'connected' like it is nowadays. The pursuit of knowledge and information is growing exponentially. This pursuit has always focused on providing the entity that engages in it with what we call these days a 'competitive advantage'. It is not an overstatement when claiming that the pursuit of a competitive advantage is at the heart of most technological advancements in the aforementioned disciplines. Competition is the keyword here. Warfare, which is the focus of this research, is essentially a competition between opposing entities where those entities actively and relentlessly seek a competitive advantage over their adversaries.

In a time and age where information technology plays a dominant role in all aspects of life, it is only natural for it to be at the center stage of efforts aimed at gaining a competitive advantage. Gaining a competitive advantage allows organizations and entities to be better-prepared to face living in a continuously changing environment. Change is the only constant in today's world. It is rapid and occurs at a pace that makes successful response to it dependent on agility. Organizations make substantial

investments in an attempt to achieve a certain degree of agility. Making your organization agile is by no means an easy task. This notion of organizational agility was discussed extensively in Alberts and Hayes (2003). Alberts and Hayes emphasized the importance of agility and went as far as saying that it will establish itself as the most consequential characteristic in the military operations of the 21st century. The key to achieving agility, however, is the ability to collect, filter, categorize, analyze, and disseminate information, because "the road to agility is paved with information" (Alberts and Hayes, 2003, 2). This poses significant challenges to command and control (C2) systems as they try to adapt their capabilities and assets to meet the thresholds that need to be attained to be ready for information age warfare.

1.1 The Power of Information

Sun Tzu, an ancient Chinese military commander, once said:

> "If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle"

It is an astonishing fact that although wars are being fought since the dawn of history (the quote above was said two thousand and a couple of hundred years ago), the value of information in warfare is ever increasing and has never taken a single step back (Coakley, 1991). Information dominance is essential to stay at the forefront, whether it is in business, technology, politics, or warfare, keeping in mind that those aforementioned areas are interconnected of-course. The world has witnessed many transitions of power from one country to another. In the 18th century, France was a principal beneficiary of

having superior infantry due to factors relating to territory, population, and agriculture in Europe (Keohane and Nye, 1998). Later on, the Naval superiority of Britain enabled it to build a global empire without using massive armies (Arquilla, 1994). Germany's power was driven by its industrial infrastructure. In current times, the United States begun its dominance by being superior in science and nuclear physics (Keohane and Nye, 1998), and is continuing that dominance today by effectively using its vast information dominance to protect its interests in the world and preserve its position as the most influential country in the world. Those instances of transitions of power are evidence that although there was a complete set of elements, with information being one of them that contributed to the dominance of nations, the full potential of information in crises and conflicts is only beginning to be realized. The Gulf War to free Kuwait from Saddam Hussein's occupation was an example of achieving the full potential of information. While the United States and its allies were amassing forces and intelligence, Saddam Hussein's forces knew next to nothing about allies' strategy. It was like a chess game where Saddam Hussein could only see his pawns, while the United States and its allies could see their pawns and those of their adversary, along with every bit of information about the environment and battlefield. The Iraqis never stood a chance.

The power of information should not be looked upon only in the "hard" sense (its use in military conflicts). The power of information also has a "soft" dimension (Nye, 2004a). Soft power is defined as "the ability to get what you want through attraction rather than coercion or payments" (Nye, 2004a, Preface). A nation's exercise of power is most effective when it makes effective use of both hard and soft power. This is how the cold war was won by the United States (Nye, 2004b). Nye so eloquently stated that due to

the information revolution in the past couple of decades, nations have never been more porous. Accordingly, the United States as a superior force in the information revolution, have successfully used the proliferation of information and American popular culture to gain attraction and understanding of American ideas and values on the global scene (Nye and Owens, 1996).

We can clearly see that information is a propelling force for any nation that is able to successfully wield its power. Not only can it win you military battles, but it can also win you the battle of hearts and minds.

1.2 All the World's a Stage: Connectivity in the Information Age

Perhaps connectivity and the diminishing boundaries between countries and people alike were the last thing on Shakespeare's mind when he wrote the poem *All the World's a Stage* (Shakespeare, 1954), but the title he selected for his poem has a profound meaning relating to the context of this subsection talking about connectivity; the world has truly become a single stage. At no point in history has the world been as "connected" as it is today. In an information technology context, borders between countries have become blurry at best, and the concept of sovereignty has been redefined. The notion of entity or individual independence has been diminishing slowly in the last couple of decades. This demise of self-sufficiency (Mulgan, 1998) is becoming an ever-present characteristic of the population today. For instance, Mulgan, pointed to the fact that modern business and corporations exhibit heavy dependency on capital markets in other continents. Products have large consumer bases in places far away from where they originated. Nations have large military bases far away from their home soil.

Harknett (1996) had the following to say about the promise of connectivity to the contemporary organization:

> "The essence of the Information Age is the emergence of a new form of organization. The information technology network seamlessly connects all of its parts, creating shared situational awareness throughout an organization. High connectivity supports both enhanced sustainability and greater accessibility."

The shared situational awareness which Harknett mentioned in the quote above is at the heart of the promise of NCO and is an important focus of this dissertation. Knowledge is a significant advantage in all aspects of life. Businesses and large corporations undertake costly research and development endeavors in order to gain the knowledge that will give them an edge over their competition. Acquiring that knowledge builds on the fact that connectivity allows the opportunity to analyze significant amounts of data about competing goods and products. In military operations, as Sun Tzu indicated, knowing oneself and the enemy is essential to victory. Therefore, connectivity on the battlefield is of utmost importance to the objective of creating the shared situational awareness that is needed to achieve effective force organization based on changes on the battlefield. Furthermore, connectivity allows communicating information about enemy force configurations, location, and assets to be seamless.

Perry et al. (2004) emphasized the importance of connectivity in military operations to achieve information superiority. They stated that information superiority is a key tenet in the U.S. military vision in building the army of the future. While they admitted that the quantitative assessment of the contribution of connectivity towards NCO is still a challenge, they argued that there is no doubt that information superiority is

consequential because it helps achieve decision dominance, new combat concepts, and crucial advantage over enemies of the future.

It is clearly evident that connectivity is the pathway to information superiority, and information superiority can be decisive on the battlefield. NCO is a military philosophy that highlights and attempts to maximize the role of connectivity in combat operations. This research will therefore attempt to study the impact of connectivity on combat force effectiveness.

1.3 Emergence of New Modes of Conflict

As previously mentioned, the vast technological development the world as witnessed in the last couple of decades has resulted in the information revolution. Like the many other areas, the way wars are fought and won was heavily influenced, and new modes of warfare have emerged. Two of the most prominent modes in the literature are cyberwar and netwar. Arquilla and Ronfeldt have authored several works that elaborate on those two concepts (See Arquilla and Ronfeldt, 1993; 1996; 1998; 2000; 2001).

They defined netwar as "information-related conflict at a grand level between nations or societies. It means trying to disrupt, damage, or modify what a target population "knows" or thinks it knows about itself and the world around it." (Arquilla and Ronfeldt, 1993, 28). An important different between netwars and other modes of conflict is that netwars focus on disrupting information and communication channels and in the most part, it is conducted using non-military activities (Whine, 1999). Netwars can be undertaken by nations, non-state actors (terrorists and organized crime organizations), or advocacy groups protesting certain government programs and policies. All those

potential netwar wagers follow a networked organization format to position themselves in the most effective way possible in a world transformed by the information revolution.

Nations around the world, especially the more tech-savvy of them, have an ever-increasing dependency on information technology as it resembles a huge part of its critical infrastructures. Increased access to the internet and media outlets around the world also make netwar deterrence a more challenging tasks as it may contain, to different degrees, a propaganda component. It is all about information. The information revolution has resulted into an unprecedented proliferation of knowledge. One of the reasons netwar has emerged is that wars are no longer entirely decided by who has the best weapons, soldiers, and jet fighters. Modern warfare has a vital information component in it as well, that is worth defending and attacking. The industrial revolution calls for a change of organizational doctrine from traditional hierarchies into interdependent networks. This change has happened and is still happening in the world. That is why netwar wagers are fundamentally networked organizations (Arquilla et al., 1999). Perhaps the most important implication of netwars is that "institutions may be defeated by networks and it may take networks to counter networks. The future may belong to whoever masters the network form" (Whine, 1999, 124).

Arquilla and Lonefeldt have also coined the concept of cyberwar. They defined it as "conducting, and preparing to conduct, military operations according to information-related principles" (Arquilla and Lonefeldt, 1993). An important distinction between netwar and cyberwar, as Arquilla and Lonefeldt defined it, is that while netwar is characterized by societal-level ideological conflict that may or may not include military operations, cyberwar happens at the military level. Cyberwar is centered on the effective

and systemic utilization of information in the implementation of military operations to neutralize and destroy adversary military forces and combat capabilities. A military force that engages in cyberwar will have its competitive advantage through its effective use of information. And contrary to earlier types of war in history, a military force undertaking cyberwarfare operations is, by design, not intended to mobilize weaponry, tactics, and systems that overwhelm the enemy both in quantity and quality, it is rather meant to produce self-synchronization between all its assets and elements before the rival force can react (Pfaltzgraff and Shultz, 1997). We again begin to get an appreciation of the importance of NCO to meet the challenges that the information age, along with its accompanying netwars and cyberwars. This appreciation will only grow larger as we begin to look more closely at historical trends in warfare and go deeper into the emerging literature that discusses the origin, building blocks, and promise of NCO.

1.4 The Vitality of Command and Control

While the lingo used in the discussion of information age warfare might have an over-emphasis on the destruction of the enemy's information collection and dissemination capabilities, this does not take away from the importance of the defensive side of information warfare. Protecting your own information capabilities in a time and age where your enemy knows that your strength lies in your knowledge-building assets is ever-more challenging. Watching the news and following media outlets whenever there is a military conflict or airstrikes that precede any conflict, we notice that the command and control capabilities are the first potential target, as weakening it makes synchronization of enemy combat assets and operations a difficult task. For example, during the Gulf War,

the Iraqi army's command and control were battered so hard and prevented Saddam's forces from keeping up with the fast pace the coalition forces were operating with. Striking the Iraqi army's command and control capabilities gave the coalition complete information dominance as well as depriving the Iraqis from any sense of situational awareness. This resulted in the Iraqi army's operations to suffer from "uncoordinated fire support, fragmented commitment of reserves, and sluggish reaction throughout the theater (Kuwait). Iraqi corps could not synchronize an adequate reaction to coalition actions even with the more capable Republican Guards" (Pardew, 1991). This gives us a sense of the vitality of command and control in information age warfare.

Alberts and Hayes (2003, 19) listed the following factors as key in adopting a correct command and control approach:

1. Warfighting environment–from static (trench warfare) to mobile (maneuver warfare).
2. Continuity of communications across echelon (from cyclic to continuous).
3. Volume and quality of information moving across echelon and function.
4. Professional competence of the decision-makers (senior officers at all levels of command) and their forces.
5. Degree of creativity and initiative the decision-makers in the force, particularly the subordinate commanders, can be expected to exercise.

By looking at those factors, specifically the first three of them, we can see how the networked organization fits into the equation of command and control. Military operations in the information age warfare are characterized by high tempo and fast pace. The command and control function therefore depends heavily on timely communication

of actionable information and thus becomes ever more vital in order to adequately respond to changing circumstances on the battlefield. Networked operations can provide this important advantage, if implemented properly, and not only can it provide an offensive advantage, but a defensive one as well (Cebrowski, 1998). After all, situational awareness, timely availability and dissemination of information across theater of operations, and synchronization of combat response and firepower, which are all provided by networked operations, are as important defensively as they are offensively.

CHAPTER 2

LITERATURE REVIEW

This chapter is dedicated to discussing the literature review and background studies related to this research.

2.1 Historical Trends in Warfare

Before we start digging deeper and deeper into information age warfare, it is very useful to look at historical trends and changes in ways wars are fought throughout history. One of the best works in this area is the book *War and Anti-War* by Alvin and Heidi Toffler. Because of their occupation as social thinkers and futurists, Toffler and Toffler (1993) linked their definitions of the transitions war has gone through to societal movements and social revolutions. Their premise was based on the fact that they believe "the way men and women make war has reflected the way they work" (Toffler and Toffler, 1993, p. 33). They described those transitions as "waves", with each emerging wave dominating and then pushing the one that preceded it aside.

First Wave War came into being with the dominance of agrarian societies which began early on in history and kept dominating until shortly before the beginning of the 18th century. Toffler and Toffler argued that there are two reasons agriculture was at the center of First Wave War:

1. Agriculture facilitated the creation and storage of economic wealth that is worth fighting for.
2. It accelerated the development of the concept of a state.

With few notable exceptions, First Wave War armies can described as being poorly organized, trained, and armed. First Wave War armies are also characterized by their irregular pay, as salaries were mostly in forms other than money as the money system was still basic and undeveloped. Communications within an army were also primitive as best and orders were given firsthand by commanders rather than being written or communicated using other means. Weapons used were designed for close hand-to-hand engagements and highly dependent on the physical strength of combatants.

Second Wave War was triggered by the industrial revolution. Shortly before the beginning of the 18th century, industrial mass production began to replace agriculture, and factories were increasingly built and developed in the West. A notable evolution took place in the loyalty of soldiers fighting Second Wave Wars. In agrarian societies, loyalty was to the landowners, tribal leaders, and warlords, but because the evolution and establishment of nation states, loyalty was given to the state. Military units were now led by professional, well-trained officers and commanders rather than warlords who although might be allies of necessity, were mostly feuding among themselves. Mass conscription was an important fixture in Second Wave Wars in times of crises and conflicts. However, the most consequential change was in standardized weapons which were developed by mass production processes. War accelerated industrialization, and standardization was soon implemented in military training and organizational doctrine. Machine guns, written communication, mechanized warfare, new military tactics were also products of Second Wave War. An example of Second Wave War dominance over First Wave War can be seen in the American civil war, where the industrialized north defeated the agricultural

south. Industrialization was at its peak in the World War II era when concepts such as mass destruction were enabled and even exercised.

It is in the late 1970s when Toffler and Toffler observed the beginnings of Third Wave War. A small group of intellectuals in the U.S. military and Congress realized a need for revitalization in the military's doctrine. This need for revitalization is rooted in the difficult experience the military went through in Vietnam. Toffler and Toffler eloquently argued that during Second Wave War, "the outer limits had already, for all practical purposes, been reached" (Toffler and Toffler, p. 43), in terms of the extension of all characteristics of weaponry, whether it is range, accuracy, or lethality.

Donn Starry, a retired U.S. army four star general, was a main protagonist in Third Wave War (Starry, 1983; 1997). Perhaps the main inspiration for General Starry's work, in addition to Toffler's book The Third Wave (Toffler, 1980), came from the October War between Israel on one side, and Egypt and Syria on the other. Specifically, certain engagements between Israel and Syria on the Golan Heights caught his attention. Syria, with more tanks and personnel, the latest weaponry the Soviet Union ever gave to an army other than its own, and the element of surprise, was unable to defeat the Israeli outnumbered and outgunned defenses, who with a simple encirclement, prevented Syrian backup troops from overwhelming their defenses and eventually fight off the attack. Starry reached the conclusion that starting ratios are irrelevant to the outcomes of war. Starry also emphasized the importance of extending the battlefield to include echelon after echelon of enemy's backup and support forces and target and encircle them to prevent them from entering battle alongside front line troops. This would require seamless communication between air and ground assets, therefore terms such as Airland

Battle and Airland Operations came into fruition. The best description of Third Wave War is in the following excerpt from Toffler and Toffler (1993, p. 68):

> "Destroy the enemy's command facilities. Take out its communications to prevent information from flowing up or down the chain of command. Take the initiative. Strike deep. Prevent the enemy's backup echelons from ever going into action. Integrate air, land, and sea operations. Synchronize combined operations. Avoid frontal attack against the adversary's strong points. Above all, know what the enemy is doing and prevent him from knowing what you are doing"

All this implies increased computerization and digitization of military assets and operations to achieve seamless synchronization of combat operations. This brings us into the concept of networked operations, which is at the heart of this research.

2.2 Network Centric Operations

Network Centric Operations/Warfare (NCO/W) has been dubbed as the most significant revolution in military affairs (RMA) in the past 200 years (Cebrowski and Garstka, 1998). Porter (2004) stated that the main reason behind the emergence of NCO is that military superiority in the 21st century is not a product of massing troops or attrition-based warfare; it is rather a product of a new information age doctrine. Porter stated that the utilization of military assets in the right place and time to accomplish a strategic advantage over your adversary is highly dependent on information sharing and connectivity.

As discussed earlier in the introduction section, NCO emerged because of the ever-increasing vitality of information in information age warfare. Miller (1998) came up with the term Information assurance (IA). He defined IA as "the security and fidelity of the information that is being passed within the myriad networked systems at multiple data

rates and security classifications." (p. 1). NCO, if implemented properly, serves as guarantee of IA.

Harknett (1996) indicated that the United States military's "Force XXI" project, which is tasked with the conception and creation of a modern 21st century military, resembles a networked form of organization. Retired General Joseph Oder, who was once the director of the military's digitization task force, defined combat force digitization on the battlefield as (Oder, 1996, p. 38):

> "the application of information technologies to acquire, exchange, and employ timely digital information throughout the Battlespace, tailored to the needs of each decider (commander), shooter, and supporter, allowing each to maintain a clear and accurate vision of the Battlespace necessary to support planning and execution."

With this excellent definition of battlefield digitization, we can begin to get an appreciation of how important it is for a combat force to possess the format of a well-connected network. This will facilitate its ability to reap the benefits of information sharing and enable speed of command and synchronization of actions on the battlefield to maximize combat effectiveness.

Following a similar theme to that of Toffler and Toffler (1993), Cebrowski and Garstka (1998) linked the evolution of warfare to societal changes and shifts in business models. They specifically stated that NCO and all of its byproducts "grow out and draw their power from the fundamental changes in American society" (p. 1). They stated that all those changes, which have been dominated by evolution in business, economics, information technology, and organizational architectures, are linked by three themes (p. 1):

1. The shift in focus from the platform to the network.

2. The shift from viewing actors as independent to viewing them as part of a continuously adapting ecosystem.
3. The importance of making strategic choices to adapt or even survive in such changing ecosystems.

For instance, they discussed how major corporations such as Wal-Mart implemented network-centric operations architectures which enabled them to achieve an information-based competitive advantage. Those architectures consist of a high-powered information backplane, sensor backplane, and transaction backplane. All of this enables the sustenance and maintenance of high levels of situational awareness with regards to the environment in which the business competition takes place. This discussion of NCO in businesses serves as an appropriate entry point to discuss how the shift to networked operations took place in the military as well.

A high-level and powerful information grid is a must for networked operations combat models. It also serves an enabler for sensor grids and engagement grids (Cebrowski and Garstka, 1998). This emphasis on information technology must not overshadow the associated leaps needed in military doctrine and organizational and training concepts. This will facilitate the sensor's grid function of achieving situational awareness and self-synchronization of combat units and assets. This will also translate into high levels of combat effectiveness as the engagement grid starts leveraging information superiority. A main tenet of NCO is the massing of effects rather than forces (Deller, 2009; Cebrowski and Garstka, 1998; Metz et al., 2006; Murdock, 2002; Hubenko et al., 2006), as we have seen in Al Qaeda's terrorist attacks on September 11, 2001 (Tranchemontagne, 2001), where fewer than thirty terrorists were able to cause more

casualties on the United States than Japan's attack on Pearl Harbor. The United States and its allies also beat Saddam's "million-man army" in the Gulf War by the massing of effects rather than forces. This was highlighted in the previous discussion about the targeting of vital Iraqi targets and command and control systems, instead of amassing allied forces for a frontal attack on Iraqi defenses and entrenchments, something that Saddam was actually hoping for. This systemic targeting of Iraqi assets was enough to cripple the Iraqi army's combat effectiveness.

As stated earlier, the crucial role information technology plays in successful implementation of NCO must not overshadow the associated leaps needed in military doctrine and organizational and training concepts. Toffler and Toffler (2003) emphasized this in their book. They stated that the majority of existing military doctrine and philosophy of command and control came into being and were cultivated during the industrial age. However, while some are still very relevant today, the manner in which the economy, information technology, and society is progressing is causing inevitable refinement and constant revision to this doctrine. For instance, industrial age mindset always adopted a "divide and conquer" approach to a wide range of problems, to the extent that military academies and organizations defined missions and roles in a very detailed and precise way (Alberts and Hayes, 2003). Alberts and Hayes also pointed out that because of this excessive specialization, different branches of industrial age armed services lack interoperability and the ability to achieve the desired degree of synchronization among the various components. Another reason for all this is centralization in planning and command and control. This all created a certain degree of rigidity in combat operations and pace.

Modern battlespace, however, calls for a more holistic and systemic approach to ensure collective situational awareness is at the highest level possible. Two key capabilities defined by Alberts and Hayes (2003) that are critical for information age militaries and therefore must take a central role in the revision of military doctrines and training programs are (1) interoperability, and (2) agility. The technological aspects that facilitate this are not the point in this discussion and were addressed previously, the human/organization side of it is. Interoperability and agility need a different form of organization to be reached. Hierarchical military structures are not suited for the fast and interdependent nature of the modern combat operations. Network-based structures allows for military configurations to be more agile and have better adaptability to the rapidly-changing and fast-paced battlespace of information age wars. The window for making decisions has become shorter and shorter as the transition from industrial to information age takes place, therefore coordination between the different branches of armed services is becoming ever-more vital. The promise of networked operations has all the necessary elements that make it a potential remedy to the dilemma of interoperability, as its emphasis on connectivity among combat assets, its more flat and agile structure, and its focus on coordination and decentralization, rather than specialization, is more suited to the type of wars that are taking place in the world today.

One of the most debated topics in the network centric operations literature is the quantification of the contribution of an information network towards the combat effectiveness of a military force. Several attempts were made to find reliable methods and metrics for quantifying a combat network performance. Notable works include those by Perry et al. (2007), Moffat (2003), Gonzalez et al. (2005), Deller (2009), Fidanci (2010).

However, some of these authors, and others (Ling et al., 2005; Fewell and Hazen, 2003), acknowledge the fact that the quantification of the contribution of information and connectivity towards combat effectiveness is still at its beginning. This research will aim to further progress the process of quantification by providing a discrete event simulation framework of the Information Age Combat Model (IACM) that also utilizes metrics Deller and Fidanci used in their work.

2.3 An Information Age Combat Model

There is often a gap between the theoretical underpinnings of a certain topic in science and the operational or engineering proficiency required to realize the promise of that theory. NCO is no different. Cares (2005) explained that the stumbling block in the way of bridging that gap is the lack of consensus on an acceptable information age combat model. Cares decided to tackle this problem and developed his vision and representation of an information age combat model. Cares' vision of an information age combat model serves as the theoretical basis for the proposed discrete event combat model proposed in this research.

Network Structure

Cares started his conceptualization of an information age combat model by discussing the structure of a network. The structure of a network refers to the definition and establishment of nodes, connections or links among these nodes, and rules for these connections. Cares repeatedly stated that the Information Age Combat Model (IACM) "should have the mathematical structure of a network" (Cares, 2005, p. 77). The basic

level of that being links that connect a set of nodes. Cares defined those nodes in the IACM as being *Sensors*, *Deciders*, *Influencers*, or *Targets*. He defined each as follows (p. 77):

- Sensors receive signals about observable phenomena in the battlespace from other nodes and send them to Deciders.
- Deciders receive information from sensors and make decisions about the present and future arrangement of other nodes.
- Influencers receive direction from Deciders and interact with other nodes to affect the state of those nodes.
- Targets are nodes that have some degree of military value but are not Sensors, Deciders, or Influencers.

All these types of nodes are connected by directional links. An instance of such a link is a Decision given by a Decider to an Influencer to engage an adversary target. An important observation regarding links in Cares' IACM is the tactical and operational nature of communications between nodes.

Combat Networks

The interactions between nodes in the IACM form what Cares termed as a combat network. Figure 1 represents the most basic form of a combat network. An important observation is the different connection types between nodes as illustrated in Figure 1. This is because each connection represents a different type of relationship depending on

the two nodes connected by that particular connection. For simplicity, this distinction will not be highlighted in future figures depicting combat network configurations.

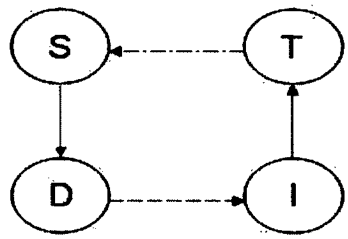

Figure 1. Simplest Form of a Combat Network

Figure 2 represents the most basic form of a combat network of two forces, with RED force shown in black and BLUE force shown in grey. This convention of RED and BLUE forces will be used in the remainder of this research.

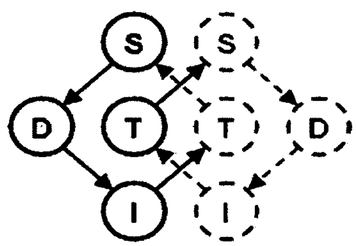

Figure 2. Simplest Combat Network of Two Forces (Deller, 2009)

Figure 3 depicts the simplest complete combat network from the definitions given by Cares. This complete network shows all the meaningful ways in which Sensors, Deciders, Influencers, and Targets interact with each other.

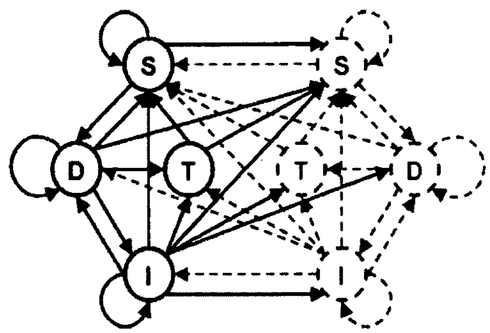

Figure 3. Simplest Complete Combat Network (Deller, 2009)

While the simplest complete combat network consisting of 8 nodes has 64 possible links, this number can be reduced by following some simple, yet important assumptions. Those assumptions were summarized by Deller (2009, p. 15):

- Targets are passive; because they are targets, they can only be sensed and influenced. Therefore, links that link Targets to nodes other than Sensors can be discarded. Accordingly, 12 links can be excluded.
- Sensors have of the task of the provision of information to Deciders and Sensors and therefore do not take actions. Consequently, 10 links from Sensors to nodes other than a Sensor or own Decider were discarded.
- Deciders carry out actions through the attached Influencers. On the other hand, they also can be sensed by opposing Sensors. Therefore, 6 links from Deciders to any enemy nodes except a Sensor were discarded.

Those assumptions enabled the reduction of link types from 64 to 28. The number can be reduced further when RED/BLUE symmetry is taken into account. This gives a total of 18 types of links. These links were tabulated by Deller (2009) and are shown in Table 1.

Cares (2005), Deller (2009), and Fidanci (2010) noted that some of those links, specifically types 1, 4, 9, 10, 11, and 13, are subject to ambiguity in their interpretation. For this reason, Deller (2009) used combat networks like the ones introduced in Figure 1, but with replacing Target nodes with enemy Sensors and Influencers. Fidanci (2010) also followed suit in his research. This research will also continue in that direction. Specifically, the implementation of the IACM in this research replaces the Target node with an enemy Sensor or Influencer. This will result in limiting the types of connections modeled to 2, 3, 6, 13, and 15. The context of the model will clarify the ambiguity of 13.

Table 1. Types of Links in the Information Age Combat Model (Deller, 2009)

Link Type	From	To	Interpretation	Link Type	From	To	Interpretation
1	S_{BLUE} S_{RED}	S_{BLUE} S_{RED}	S detecting own S, or S coordinating with own S	10	I_{BLUE} I_{RED}	D_{BLUE} D_{RED}	I attacking own D, or I reporting to own D
2	S_{BLUE} S_{RED}	D_{BLUE} D_{RED}	S reporting to own D	11	I_{BLUE} I_{RED}	I_{BLUE} I_{RED}	I attacking own I, or I coordinating with own I
3	S_{BLUE} S_{RED}	S_{RED} S_{BLUE}	S detecting adversary S	12	I_{BLUE} I_{RED}	T_{BLUE} T_{RED}	I attacking own T
4	D_{BLUE} D_{RED}	S_{BLUE} S_{RED}	S detecting own D, or D commanding own S	13	I_{BLUE} I_{RED}	S_{RED} S_{BLUE}	I attacking adversary S, or S detecting adversary I
5	D_{BLUE} D_{RED}	D_{BLUE} D_{RED}	D commanding own D	14	I_{BLUE} I_{RED}	D_{RED} D_{BLUE}	I attacking adversary D
6	D_{BLUE} D_{RED}	I_{BLUE} I_{RED}	D commanding own I	15	I_{BLUE} I_{RED}	I_{RED} I_{BLUE}	I attacking adversary I
7	D_{BLUE} D_{RED}	T_{BLUE} T_{RED}	D commanding own T	16	I_{BLUE} I_{RED}	T_{RED} T_{BLUE}	I attacking adversary T
8	D_{BLUE} D_{RED}	S_{RED} S_{BLUE}	S detecting adversary D	17	T_{BLUE} T_{RED}	S_{BLUE} S_{RED}	S detecting own T
9	I_{BLUE} I_{RED}	S_{BLUE} S_{RED}	I attacking own S, or S detecting own I	18	T_{BLUE} T_{RED}	S_{RED} S_{BLUE}	S detecting adversary T

Perron-Frobenius Eigenvalue

The IACM can use matrix representations (Cares, 2005). A description of a combat network as a graph is provided by an adjacency matrix A. The rows and columns

of adjacency matrices represent the different nodes. Matrix elements are either a 0 or 1, depending on the existence of a connection between a particular set of two nodes. 0 means no such connection exists, while 1 indicates a connection. For a particular force to be combat effective, connected cycles must be present that include the node to be influenced; the target node. This attention to cycles is attributed to the work of Cares (2005). Cares stated that there is a relationship between the existence of cycles and the ability of Forces to display increased networked effects. Deller (2009) defined those cycles as sub-networks or "arrangements of linked nodes where the path of directional links revisits at least one node previously departed from" (p. 17).

One of the metrics that can be calculated from adjacency matrices is the eigenvalue (λ_{PFE}). It was proved that for a graph that does not include any closed cycles, $\lambda_{PFE} = 0$ (Jain and Krishna, 1999). Moreover, a graph, or network in the context of this research, that has a single cycle of a particular length, $\lambda_{PFE} = 1$. Therefore, Cares (2005) proposed the adoption of λ_{PFE} as a metric to measure the ability of a network to exhibit networked effects, and therefore, increased combat effectiveness. Accordingly, Deller (2009) and Fidanci (2010) used this metric in their work. Figure 4 shows the adjacency matrix for the simplest complete combat network.

	S	D	I	T	S	D	I	T
S	1	1	0	0	1	0	0	0
D	1	1	1	1	1	0	0	0
I	1	1	1	1	1	1	1	1
T	1	0	0	0	1	0	0	0
S	1	0	0	0	1	1	0	0
D	1	0	0	0	1	1	1	1
I	1	1	1	1	1	1	1	1
T	1	0	0	0	1	0	0	0

Figure 4. Adjacency Matrix for the Simplest Complete Combat Network (Fidanci, 2010)

Although the eigenvalue was strongly emphasized by Cares, he also proposed several other metrics that measure the network's performance with regards to several important considerations such as adaptability, survivability, robustness, and many other properties that are desirable to maintain in a combat network. Those metrics include (Cares, 2005):

- Number of nodes (N): the presence of networked effects depends on the existence of a fairly large number of nodes.
- Link to node ratio (l/N): This ratio is dependent on whether the network is maximally connected, where every node has a direct connection to every other node (l = (N – 1)!, or minimally connected, where nodes are connected with the minimum number of links possible.
- Degree Distribution: Links should not be uniformly-distributed for adaptive network performance. The degree of a node means the number of links directly connected to a particular node. Cares stated that Adaptive, complex networks have a skew distribution, where there are a very small number of nodes that are highly connected, a moderate number of nodes that are moderately connected, and a large number of nodes that are minimally-connected.
- Size, connectivity of the largest hubs: damage to any of the small number of large, well-connected hubs can quickly propagate to other large hubs in a complex network. Therefore, Cares stated that combat networks should be engineered in a manner that ensures the largest hubs possess no direction connections.

- Characteristic Path Length: Cares defined this metric as the middle ranked value (i.e. median) of the mean of the lengths of all shortest paths in the network. He indicated that this value grows by the order of the magnitude of the number of nodes in the combat network.
- Clustering Coefficient: This is primarily a measure of network cohesion and self-synchronization. It is "the proportion of a node's direct neighbors that are also direct neighbors of each other" (p. 103).
- Betweenness: This is a measure of a node's significance to entire network structure. Specifically, it measures the proportion of shortest paths that pass through a node and therefore identifies potential bottlenecks in the flow of a network.
- Path horizon: a measure of the number of nodes on average in a network that a node must interact with for consecutive self-synchronization to occur.
- Neutrality rating: the additional structure in a complex adaptive network over the minimum requirements of connectivity.
- Coefficient of networked effects (λ_{PFE}/N): a measure of the amount of cyclic behavior per each node. This measure allows a comparison between networks of varying sizes with regards to the potential for networked effects.
- Susceptibility: a metric that measures the number of links or nodes that can be removed from the network before its dynamic structure begins to break down.

Deller (2009) utilized the eigenvalue and all the aforementioned metrics proposed by Cares in his work. Deller found that the eigenvalue (and its dependent coefficient of networked effects) was the only metric that varied in value. Therefore, it was the only metric that might be of importance in understanding the variability in combat effectiveness between different combat configurations. However, Deller noted that the utility of λ_{PFE} as a discriminating factor between configurations diminishes as the number of unique combinations gets larger. This conclusion was also echoed by Fidanci (2010) as well. Therefore, Deller introduced two additional metrics, Disparity and Robustness, in an attempt to gain more insight into the relationship between the extent of networked effects and combat effectiveness. Fidanci also used those two metrics and suggested additional ones and ran regression analysis on his results. All those metrics, except the λ_{PFE} due to its diminishing importance, were adopted by this research and they are discussed thoroughly in Chapter 3.

Combat Cycles

The concept is combat cycles is of high importance in this research. A combat cycle is defined as the set of interactions that enables a particular combat network to influence an adversary node, therefore, destroying it or rendering it combat ineffective. In the context of this research, a combat cycle consists of communication and coordination between a Sensor, a Decider, and Influencer, aimed towards an eventual attack on and elimination of an enemy target. A combat cycle starts with a Sensor sensing an enemy target, with this target being an enemy Sensor or Influencer. The Sensor communicates

information relating to the location and nature of the enemy target to its Decider. The Decider then instructs an Influencer to take out that target.

It is important to realize the meaning of a cycle in this context. Specifically, it takes a set of connected Sensor-Decider-Influencer to form a combat cycle. For instance, a Decider that has any number of Sensors but no Influencers cannot form a combat cycle. Similarly, a Decider that has Influencers assigned to it but all of its Sensors were eliminated is also incapable of generating a combat cycle. Therefore, the presence of a connection between those three types of nodes is a prerequisite to generating combat cycles.

Results of Previous IACM Modeling Endeavors

The work of Deller (2009) and Fidanci (2010) represent previous attempts at modeling the IACM based, on the most part, on how Cares (2005) envisioned it. Deller and Fidanci used agent-based modeling as their modeling paradigm. The output of the simulation focused on the probability of win for a force over another, namely BLUE and RED. The probability of win for a BLUE force combination is defined as the percentage of BLUE wins obtained for that particular combination from the total number of engagements conducted for it.

Deller used NetLogo as the implementation tool for his agent-based simulation. NetLogo was developed was Northwestern University's Center for Connected Learning and Computer-Based Modeling. For more information on NetLogo and how it is used in modeling and simulation, see Wilenski, 1999; Earnest, 2008; Damaceanu, 2008; Khasawneh et al., 2009.

Due to the high computational requirements of Deller's model and the classification of his work as a first step into modeling the IACM, he ran a limited number of combat configurations. An example of these complex computational demands is the fact that it took Deller 78 hours to run all the unique combinations of one configuration, the 9-5-9 force.

Deller ran several configurations and conducted regression analysis on his results to try to determine the predictive power the λ_{PFE} wields over the probability of winning a certain engagement. The λ_{PFE} was found to be a significant factor for combat configurations with a small number of Deciders. For instance, the regression models for a 7-3-7 and 8-3-8 configurations had coefficients of determination (R^2) of 89%, and 87%. However, Deller noted that the utility of λ_{PFE} as a discriminating factor between configurations diminishes as the number of unique combinations gets larger, and specifically, as the number of Deciders increases. For instance, the 9-5-9 configuration yielded an R^2 of only 51%. This effectively means that the percentage of variation in the probability of a win explained by the λ_{PFE} was only 51%. Therefore, other factors are responsible for 49% of variation.

Deller added two additional metrics, Disparity and Robustness, to augment the effectiveness of λ_{PFE} as a metric of networked performance of configurations. The coefficient of determination (R^2) improved when, for instance, he added Robustness to the mix. It increased to 80%. It can be clearly seen how the addition of other metrics is crucial to have a more generalized understanding of the factors affecting combat effectiveness.

Fidanci (2010) also used agent-based modeling, but he utilized a different implementation platform in the form of AnyLogic. AnyLogic is a multi-paradigm simulation tool developed by XJ Technologies. For more works on how AnyLogic is used in agent-based modeling, refer to Wang et al., 2008; Shendarkar et al., 2008; and Siebers et al., 2010.

Fidanci expanded on the work of Deller by running more configurations and proposing additional metrics. Specifically, he ran 55 experiments and proposed four additional metrics, which are Strength, Power, Stability, and Connectivity. Fidanci defined each of those metrics and gave a formula to calculate each. Those metrics are discussed in detail in Chapter 3. Fidanci's model also had "heavy" computational demands. It took him around 3 months, while using several machines, to collect all his data.

Fidanci's echoed similar conclusions to the work of Deller (2009) with regards to the utility of the λ_{PFE} as a metric for networked performance. He found that the λ_{PFE} was a fair predictor for configurations that possessed up to seven deciders. When Fidanci added Deller's metrics of Disparity and Robustness, the R^2 increased substantially to around 79%. Fidanci's own metrics, Power and Connectivity combined with the λ_{PFE}, put the R^2 value at around 80%.

It is important to emphasize the fact that the work of Deller and Fidanci focused solely on symmetric force sizes engaging in symmetric warfare. Specifically, their model simulated battle between forces that have an equal number of Sensors and Influencers, thus maintaining an X-Y-X configuration, with X representing the number of Sensors and Influencers, and Y representing the number of Deciders. Those configurations did battle

against configurations with identical sizes (i.e. symmetric warfare). The results of the work of Deller and Fidanci indicate that organizing forces for networked effects, in the case of balanced force sizes engaged in symmetric warfare, does offer an advantage in combat operations over adversaries. Furthermore, they were also able to suggest quantifiable metrics that offer a fair degree of predictive power with regards to the outcome of combat engagements.

This research will take this to the next level by simulating unbalanced force configurations in which the number of Sensors and Influencers varies across configurations (i.e. X-Y-Z). This will help in understanding how networked effects influence the outcome of combat engagements between two forces which possess unbalanced sizes. Moreover, this research will also attempt to pinpoint the significant metrics and factors of interest in determining the outcome of combat engagements. This will aid in the design and engineering of combat network to ensure they are organized and arranged in a way that maximizes their potential for networked effects, and consequently, combat effectiveness. Unbalanced force configurations will be simulated engaging in symmetric warfare.

In contrast to previous attempts at modeling Cares' IACM, which utilized the agent-based simulation paradigm, this research decided to take a different path for a variety of reasons. Discrete event simulation was identified as a more appropriate paradigm for the implementation of the IACM. More information on this matter and a detailed comparison between the agent-based simulation as Deller and Fidanci envisioned it, and the discrete-event simulation alternative as this research envisions it, is given in Chapter 3. For information than can aid in providing a general understanding of the

discrete event simulation paradigm, interested readers are referred to Pooch and Wall, 1993; Fishman, 2001; Robinson, 2004; and Banks, 2005.

CHAPTER 3

RESEARCH METHODOLOGY AND FRAMEWORK

This chapter will be dedicated to discussing the methodology and framework followed in this research. It will start by giving a generalized, high-level, seven-step articulation of the methodology followed. The rest of this chapter will be dedicated to give a more detailed articulation of the research methodology and the sequence of steps and events that were carried out to execute this methodology.

3.1 High-Level Articulation of Research Methodology

The methodology followed in this research followed the general guidelines provided by those seven steps:

1. Problem Identification: The research gap is the lack of work dedicated towards the quantification of the contribution of the arrangement of networked forces towards their combat effectiveness in the case of unbalanced combat configurations engaging in symmetric warfare. Moreover, previous modeling attempts displayed moderate running speed and efficiency. Therefore, an alternative approach with improved performance is needed. This research will build on the work of Deller (2009) and Fidanci (2010) towards those ends.
2. Review of Literature and Background Studies: This research will develop an understanding of the subject area by an in-depth look at relevant studies in the literature regarding network centric operations. Emphasis will be placed on the work of Cares (2005), Deller (2009) and Fidanci (2010).

3. Research Objective: The objective is to identify indicators/predictors that are significant in determining the contribution of the arrangement of forces towards its overall combat effectiveness, with focus on forces of unbalanced sizes engaging in symmetric warfare. Moreover, this research will show how the proposed discrete event combat model offers higher performance levels in terms of speed and efficiency. Therefore, it will be a viable as well as more efficient alternative to agent-based models previously developed by Deller (2009) and Fidanci (2010).
4. Research Design: The proposed combat model will run all configurations of unbalanced sizes, for values of X, Y, and Z $\leq$ 11. Those unbalanced configurations will be simulated doing symmetric combat engagements. Specifically, each X-Y-Z will do battle against an identical X-Y-Z force.
5. Run Experiment: Each battle engagement will be run 30 times (i.e. replications).
6. Data Generation, Collection, and Analysis: several Visual Basic programs were written to extract data from the huge output files and calculate the performance metrics. Linear and nonlinear regression analysis will be carried out to assess the predictive power of the metrics chosen.
7. Conclusions and Future Work: Based on the results of the data analysis, conclusions will be made about the utility of the performance metrics selected in the quantification of the impact of networked operations on combat effectiveness. Furthermore, suggestions about future relevant research work will be given.

3.2 Partitioning in Mathematica

One of the most important steps in this research is the manner in which Sensors and Influencers are distributed across Deciders. Regardless of the values of X-Y-Z, there is always a finite number of combinations that can be used to arrange those assets. Guided by the assumption that each Sensor and Influencer can only be attached to one Decider and that Deciders must be connected to one or more of Sensors and Influencers, it can be clearly seen how this turns into an integer-partition problem.

In number theory, an integer partition of an integer n is a way of writing n as a sum of positive integers. In the context of this research, the order of those integers is not important. What is of concern is the number of unique or meaningful combinations. As an example, look at Table 2 below. This table shows the number 8 partitioned 3 at a time.

Table 2. The Number 8 Partitioned 3 at a Time

Partition	Partitions of the Number 8		
1	6	1	1
2	5	2	1
3	4	3	1
4	4	2	2
5	3	3	2

Note that the 6-1-1 partition is the same as 1-6-1 or 1-1-6. The latter two are considered redundant, and are not included in the Table 2. Similarly, when partitioning Sensors and Influencers, only those partitions that are unique are of interest to this research. The redundant ones are discarded from the simulation. More information about what is a unique combination in the context of this research can be found in Appendix A.

Mathematica® was used to help with partitioning. The Mathematica command that yields the result in Table 2 is:

```
IntegerPartitions[8,{3}]
```

The code fragment that displays the output in tabular format is:

```
TableForm[IntegerPartitions[8,{3}]]
```

The code fragment that counts the number of unique partitions is:

```
Length[IntegerPartitions[8,{3}]]
```

This yields an answer of 5. Those 5 unique combinations are actually the same shown in Table 2. The code that determines the number of permutations for each partition configuration is:

```
TableForm[Permutations[IntegerPartitions[8,{3}][[k]]]]
```

Where *k* it the *k*th item in the list. Take this code fragment as an example:

```
TableForm[Permutations[IntegerPartitions[8,{3}][[1]]]]
```

This yields all the permutations of the first partition of 8, which was seen earlier to be 6-1-1. The output is shown in Table 3.

Table 3. All Permutations for the First Partition (k = 1)

6	1	1
1	6	1
1	1	6

Take the following code fragment as another example:

```
TableForm[Permutations[IntegerPartitions[8,{3}][[2]]]]
```

This will yield all the possible permutations of the second partition, which is 5-2-1. The output is shown in Table 4.

Table 4. All Permutations for the Second Partition (k = 2)

5	2	1
5	1	2
2	5	1
2	1	5
1	5	2
1	2	5

The maximum value for k for an 8-3 partition is 5 because that is the total number unique partitions for the problem.

In order to determine all of the unique combat configurations for all Sensors, Deciders, and Influencers, all of the permutations for the Sensors and Deciders; and Deciders and Influencers must first be determined. Because the number of deciders for each combat cycle is always one, essentially the problem reduces to simply the permutations for the Sensors. The same technique used to determine the permutations for the Sensors can be used to determine the permutations for the Influencers. Therefore, we used Mathematica to produce a number of data files that contain the number of permutations for each X-Y partition. These data files are then used as the input for the Visual Basic program that was written to program the technique described by Fidanci (2010) to determine and output the number of unique combat configurations.

Based on the minimum and maximum values selected for this research for the X-Y-Z configurations, which are 3 and 11, respectively, Mathematica was used to

determine the total number of permutations for $3 \leq x \leq 11$ and $3 \leq y \leq 11$ to produce those data files for the X-Y partitions. This code is shown below:

```
For[i=3,i<=20,i++,
For[j=3,j<=i,j++,
Export["D:/Partitions/Partitions_"<>IntegerString[i]<>"_"<>I
ntegerString[j]<>".dat",TableForm[Flatten[Table[Permutations
[IntegerPartitions[i,{j}][[k]]],{k,1,Length[IntegerPartition
s[i,{j}]]}],1]]]]]
```

Therefore, what this code does is create X-Y partition files that have a ".dat" extension for each X-Y partition, for all X-Y values between 3 and 11. Those files will contain all the possible combinations, even redundant ones, for that particular X-Y partition. The code will loop through all partitions starting from 3-3 and ending with 11-11 and create files for each one.

3.3 Determination of Unique Combinations

Determining the number of unique combinations for each configuration is one of the most challenging tasks when it comes to modeling the information age combat model. Deller (2009) stated that this step is absolutely critical for attempts aimed at scoping this problem and is not a trivial task. It is also essential for computational reasons. For instance, the 7-3-7 configuration has 225 permutations/combinations. Asking the Combat Model to run the simulation of a BLUE 7-3-7 configuration fighting a RED 7-3-7 configuration, would require 225^2 engagements, which is 50,625. On the other hand, the 7-3-7 configuration has only 42 unique combinations. This means that 183 combinations or 33,489 simulation engagements are simply redundant and running them will only waste time and computer resources and would constitute a very inefficient Combat Model. Consequently, this means that the data resulting from those runs does not possess

any value in the analysis. Forty-two unique combinations would require 42^2 engagements, which give a much downsized 1,764 engagements, and therefore constitute a much more efficient Combat Model. As the values become larger and larger, we can begin to understand how a significant amount of time and computer resources can be saved when the simulation experiments needed for this research are run. Appendix A presents a brief explanation of unique combinations and how they differ from redundant combinations. Fidanci (2010), who continued and expanded the work by Deller (2009), tackled this problem. The technique implemented in this research, and programmed in Visual Basic, to determine the unique combinations for each configuration was developed by him.

The way his technique works is by using a special matrix operation. Take the 5-3-5 configuration for instance. The 5-3 partitions output from Mathematica is shown in Table 5 below. This table shows the number of Sensors and Influencers (S, I) assigned to each one of the three Deciders (D_n). For instance, the first row indicates that three Sensors and three Influencers are assigned to D_1, while D_2 and D_3 have one of each. Those combinations are not all unique as can be easily observed, some are redundant.

Table 5. Mathematica Output for 5-3 Partitions

S, I for D1	S, I for D2	S, I for D3
3	1	1
2	2	1
2	1	2
1	3	1
1	2	2
1	1	3

Fidanci arranged those combinations into a matrix, A, and then calculated the transpose of that matrix, A^T. A is used to represent the 5-3 partitions for distribution of

Sensors across Deciders, and A^T is used to represent the 5-3 partitions for Influencers across Deciders. After determining A and A^T, a special operation is applied to those two matrices. A and A^T are shown in Figure 5.

$$\begin{vmatrix} 3 & 1 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 3 \end{vmatrix} \odot \begin{vmatrix} 3 & 2 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 3 & 2 & 1 \\ 1 & 1 & 2 & 1 & 2 & 3 \end{vmatrix}$$

Figure 5. A and AT for 5-3 Partitions

The manner in which this special operation works is similar, but not exactly identical to normal matrix multiplication. Specifically, it is similar to multiplication in terms of certain aspects of the process as well as in determining the size of the resulting matrix. However, instead of actually multiplying a matrix element by another, another mathematical operation is applied. This operation can be expressed as (Fidanci, 2010):

$$\mathbf{AoA'} = \begin{bmatrix} a_{11} & \cdots & a_{1y} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{my} \end{bmatrix} \mathbf{o} \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{y1} & \cdots & a_{ym} \end{bmatrix} = \sum_{j=1}^{y} \frac{a_{ij}^{y}}{a_{ji}^{1/y}}$$

Where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1y} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{my} \end{bmatrix}, \quad \mathbf{A'} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{y1} & \cdots & a_{ym} \end{bmatrix}, \text{ and:}$$

$1 \le i \le m$, and y = Decider Number

Figure 6 illustrates how this operation is applied and displays the resulting matrix.

$$
= \begin{bmatrix}
\frac{3^z}{3^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{3^z}{2^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{3^z}{2^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{3^z}{1^{1/z}}+\frac{1^z}{3^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{3^z}{1^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{3^z}{1^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{1^z}{3^{1/z}} \\
\frac{2^z}{3^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{2^z}{2^{1/z}}+\frac{2^z}{2^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{2^z}{2^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{2^z}{3^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{2^z}{2^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{1^z}{3^{1/z}} \\
\frac{2^z}{3^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{2^z}{2^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{2^z}{2^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{2^z}{2^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{1^z}{3^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{2^z}{2^{1/z}} & \frac{2^z}{1^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{2^z}{3^{1/z}} \\
\frac{1^z}{3^{1/z}}+\frac{3^z}{1^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{3^z}{2^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{3^z}{1^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{3^z}{3^{1/z}}+\frac{1^z}{1^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{3^z}{2^{1/z}}+\frac{1^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{3^z}{1^{1/z}}+\frac{1^z}{3^{1/z}} \\
\frac{1^z}{3^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{2^z}{2^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{2^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{2^z}{3^{1/z}}+\frac{2^z}{1^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{2^z}{2^{1/z}}+\frac{2^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{2^z}{1^{1/z}}+\frac{2^z}{3^{1/z}} \\
\frac{1^z}{3^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{3^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{3^z}{1^{1/z}} & \frac{1^z}{2^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{3^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{1^z}{3^{1/z}}+\frac{3^z}{1^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{1^z}{2^{1/z}}+\frac{3^z}{2^{1/z}} & \frac{1^z}{1^{1/z}}+\frac{1^z}{1^{1/z}}+\frac{3^z}{3^{1/z}}
\end{bmatrix}
$$

$$
= \begin{bmatrix}
20.7208 & 23.2236 & 23.2236 & 28.6934 & 28.5874 & 28.6934 \\
14.5469 & 13.6992 & 15.1433 & 14.5469 & 15.1433 & 16.6934 \\
14.5469 & 15.1433 & 13.6992 & 16.6934 & 15.1433 & 14.5469 \\
28.6934 & 23.2236 & 28.5874 & 20.7208 & 23.2236 & 28.6934 \\
16.6934 & 15.1433 & 15.1433 & 14.5469 & 13.6992 & 14.5469 \\
28.6934 & 28.5874 & 23.2236 & 28.6934 & 23.2236 & 20.7208
\end{bmatrix}
$$

Figure 6. Special Matrix Operation (Fidanci, 2010)

Now that we have the resulting matrix, the logic becomes simple. Just count the number of matrix elements while ignoring redundant instances of the same number. The total will give the number of unique combinations for that particular configuration. For instance, there are six instances of the number "23.2236" in the resulting matrix, as shown in Figure 7. This means that the constituting combination behind each instance is identical (i.e. redundant), therefore, we can use one instance of that combination and discard the remaining ones.

Take the first and second instance for example. The constituting combination behind the first instance is 3, 1, 1 for Sensors and 2, 2, 1 and Influencers. The second instance includes 3, 1, 1 for Sensors and 2, 1, 2 for Influencers. Figure 8 depicts those two combinations. It is clear, based on the explanation given in Appendix A that those two combinations are redundant. Therefore, in the case of the 5-3-5 configuration,

counting the elements of the resulting matrix in Figure 7 while discarding redundant values, gives a total of 8. Therefore, there are a total of 8 unique combinations for a 5-3-5 configuration.

$$= \begin{bmatrix} 20.7208 & \boxed{23.2236} & \boxed{23.2236} & 28.6934 & 28.5874 & 28.6934 \\ 14.5469 & 13.6992 & 15.1433 & 14.5469 & 15.1433 & 16.6934 \\ 14.5469 & 15.1433 & 13.6992 & 16.6934 & 15.1433 & 14.5469 \\ 28.6934 & \boxed{23.2236} & 28.5874 & 20.7208 & \boxed{23.2236} & 28.6934 \\ 16.6934 & 15.1433 & 15.1433 & 14.5469 & 13.6992 & 14.5469 \\ 28.6934 & 28.5874 & \boxed{23.2236} & 28.6934 & \boxed{23.2236} & 20.7208 \end{bmatrix}$$

Figure 7. Redundancy in Matrix Elements

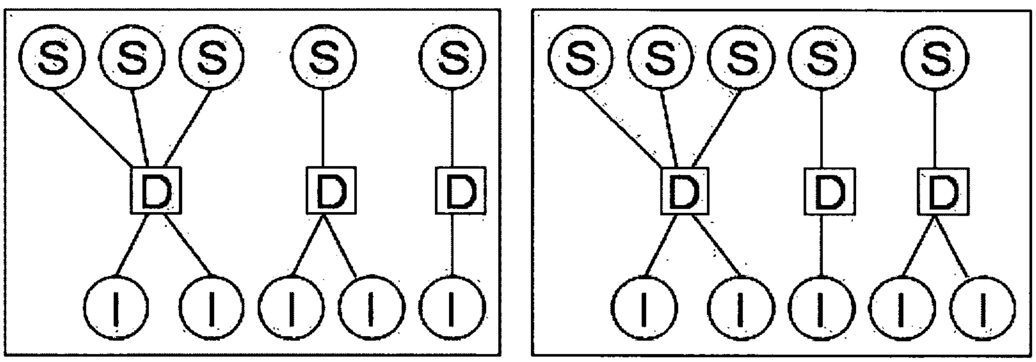

Figure 8. Two Redundant 5-3-5 Combinations

A Visual Basic Program was written to streamline and automate the application of Fidanci's technique to all the partition files Mathematica produced. X-Y partition files were combined using Fidanci's technique and data files were produced. Each data file contains the unique combinations for a particular X-Y-Z configuration. This program also lists all configurations and counts the number of unique combinations for each configuration. The code for this program is shown in Appendix B. The data files that this

program produces, which contain the unique combinations for each configuration, are used as input to the proposed discrete event Combat Model.

3.4 Unbalanced Configurations for Symmetric Engagements

The maximum number of simulations that were determined to be reasonable using current computer technology is for configurations of $3 \leq X$, Y, and $Z \leq 11$. Since the combat model developed in this research does symmetric engagements, the model would have to simulate battle between a RED X-Y-Z configuration against a BLUE X-Y-Z configuration of the same size. This gives an overall total of 285 configurations. Therefore, a Visual Basic program was written to count the total number of configurations and list them. The code for this program is shown in Appendix C. Another Visual Basic program was written to calculate the number of unique combinations associated with each configuration, as well as the number of unique symmetric engagements for $3 \leq X$, Y, and $Z \leq 11$. The code for this program is shown in Appendix D. All 285 configurations ran in this research and the number of unique combinations asociated with each are shown in Table 14 in Appendix D. Table 6 below shows the number of unique engagements for configurations at different intervals. The second column is cumulative. Figure 9 depicts the number of unique engagements. The last number in Table 6, which is around 4 million, is the number of unique engagements conducted in this research. Each engagement was ran for 1,000 replications.

Table 6. Unique Symmetric Engagements for Y ≤ X and Z ≤ X and 3 ≤ X ≤ 11

S-D-I	Number of Configurations	Number of Unique Engagements
3	1	1
4	5	8
5	14	119
6	30	986
7	55	6,583
8	91	36,089
9	140	179,032
10	204	834,621
11	285	3,724,225

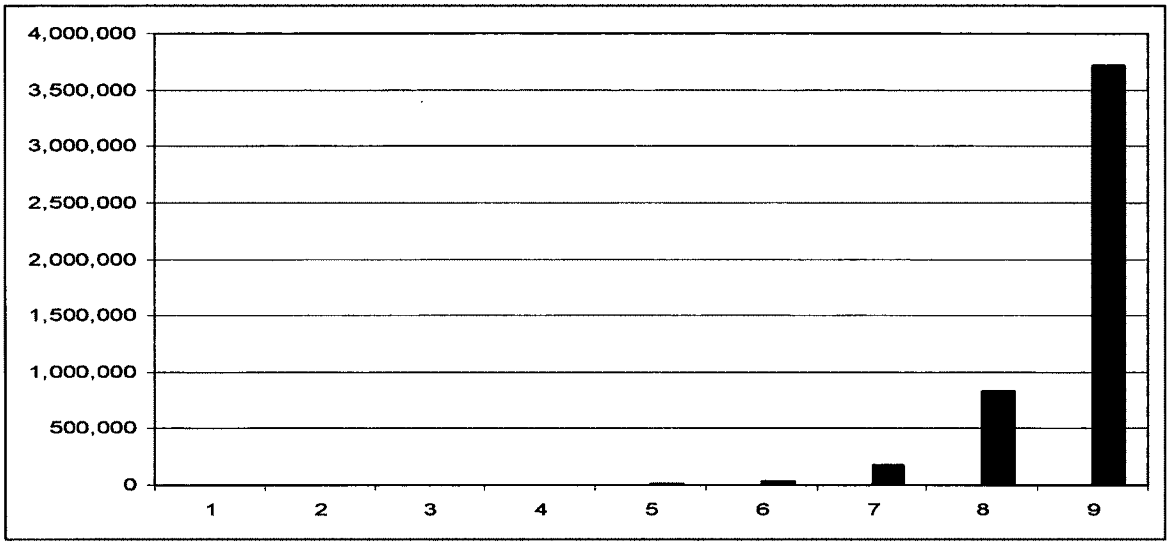

Figure 9. Unique Symmetric Engagements for Y ≤ X and Z ≤ X and 3 ≤ X ≤ 13

Initially, the decision was to go to 15 levels instead of 11, which gives a total of 819 configurations instead of 285. However, it was determined that the number of unique engagements will become overwhelming in terms of current computer processor technology. Table 7 shows the number of unique symmetric combat engagements in the case of increasing the scope of this research to X, Y, Z values of 15. Figure 10 depicts the number of unique engagements.

It is noticed that the number of unique engagements increased significantly because of the addition of four levels (from 11 to 15). It was close to 4 million engagements for 11 levels, but it exceeded the 1 billion mark for 15 levels. We can begin to understand how this becomes prohibitive with current computer processor technology. Thus far, all those numbers were given for the symmetric case.

Table 7. Unique Symmetric Engagements for Y ≤ X and Z ≤ X and 3 ≤ X ≤ 15

S-D-I	Number of Configurations	Number of Unique Engagements
3	1	1
4	5	8
5	14	119
6	30	986
7	55	6,583
8	91	36,089
9	140	179,032
10	204	834,621
11	285	3,724,225
12	385	16,103,308
13	506	67,851,052
14	650	279,547,570
15	819	1,128,154,812

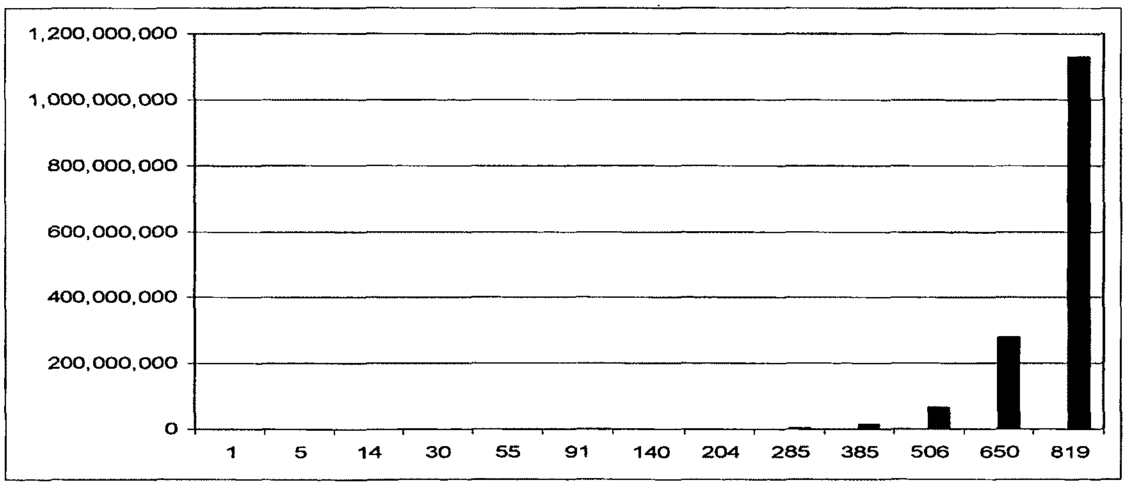

Figure 10. Unique Symmetric Engagements for Y ≤ X and Z ≤ X and 3 ≤ X ≤ 15

Imagine the computational resources required to run 819 configurations in the case of asymmetric warfare, where each configuration does battle against all other configurations. Another Visual Basic program was written to determine the number of unique engagements in the case of asymmetric warfare. This program is shown in Appendix E. Results are shown in Table 8 and Figure 11. The number of unique engagements that would be required is now a 67+ billion. Therefore, in addition to restricting levels to 11, the scope of this research was restricted to symmetric warfare.

Table 8. Unique Asymmetric Engagements for $Y \leq X$ and $Z \leq X$ and $3 \leq X \leq 15$

S-D-I	Number of Configurations	Number of Unique Engagements
3	1	1
4	5	22
5	14	540
6	30	7,455
7	55	75,872
8	91	604,489
9	140	4,093,966
10	204	24,506,051
11	285	133,909,613
12	385	682,010,552
13	506	3,288,234,014
14	650	15,167,658,763
15	819	67,445,488,254

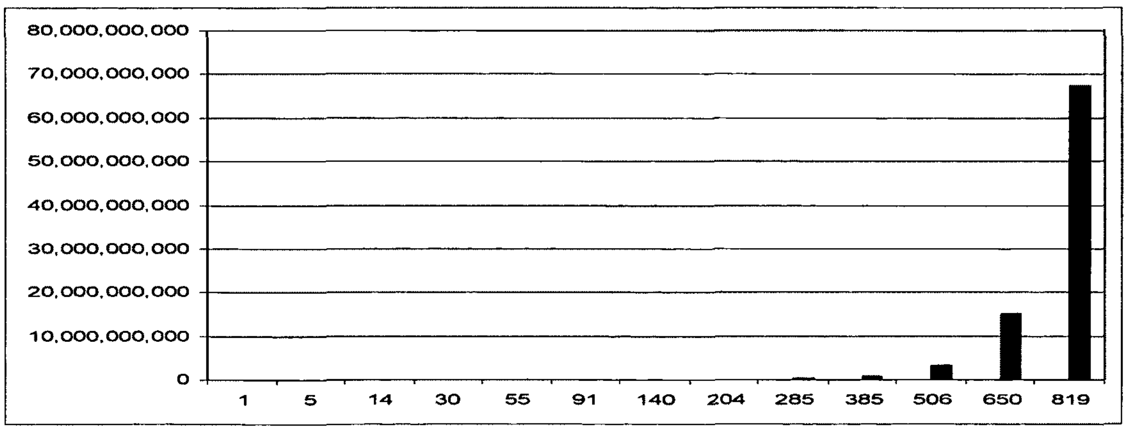

Figure 11. Unique Asymmetric Engagements for $Y \leq X$ and $Z \leq X$ and $3 \leq X \leq 15$

3.5 Discrete Event Simulation: Direct Combat for More Speed and Efficiency

Previous attempts at modeling Cares' IACM utilized agent-directed simulations. In contrast, this research reached the determination that discrete event simulation also represents a viable, and even better, option to model the IACM. Better in this context means more efficiency and speed of simulation. Before comparing the approach this research adopted to that of Deller and Fidanci, a brief introduction of discrete event simulation will be given.

Perhaps one of the best definitions given for discrete event simulation was the definition articulated by Robinson (2004). Robinson defined it as a simulation paradigm that gives the operation of a system a representation as a chronological succession of events. Specifically, Robinson stated that "in discrete-event simulation only the points in time at which the state of the system changes are represented" (p. 15). This definition touches the heart of the approach adopted in this research. The reason is that the rationale and motivation behind utilizing discrete event simulation is pretty simple and can be expressed in two words; *direct combat.*

One of the reasons behind the demanding agent-based simulation environment which Deller (2009) and Fidanci (2010) used, in terms of computer processor technology, is that a lot of processing capabilities are needed and utilized to simulate the movement of agents across the simulation space. Keeping track of the location of each agent with regards to other agent and consequent use of this information to program the "sensing" function of sensors is not a trivial task in terms of processing requirements. The Sensing rule sets, sensing range implementation, and transfer of information to Decider agents, are all complex procedures. Furthermore, both the determination of the nearest target

node that was sensed and the dispatching of an Influencer to eliminate that target are also activities that have no added-value from a discrete event simulation standpoint. Specifically, following the rationale that Robinson coined, all those activities reflect a representation of the system at times where no actual change in the state of the system took place. Therefore, those simulation activities only serve to "clog up" the computer's processors and burden it with additional processing requirements that are non-essential to the purpose of the simulation. A more direct approach of combat was therefore conceptualized and implemented in this research to reap the benefits which discrete event simulation offers in this particular modeling proposition, in terms of increased efficiency and speed of simulation. This was necessary because of the increase in number of configurations tested compared to previous work that modeled the IACM. Specifically, Deller's work was an introduction and a first-cut effort, and his focus was more on the quantification of the impact of networked effects on combat effectiveness. Therefore, he ran a handful of experiments. Fidanci's agent-based simulation was an expansion of both the quantification side and the modeling side, and he ran a total of 55 configurations, that required around 3 months of simulation work. The design and focus area of this research requires 285 configurations, therefore, the need for more efficient alternative cannot be stressed enough.

Modeling the points in time at which a change in the state of the system takes place was the guiding principle in developing the proposed discrete event combat model in this research. Visual Basic 6.0 was selected as the implementation platform. Familiarity with the Visual Basic programming language was not the only reason behind its selection. Visual Basic is considered a very flexible environment in terms of the ease

and speed at which files are read from for input, and written to for output. Furthermore, the data generation process as well as calculation of metrics can be easily automated and streamlined by writing several Visual Basic program that interface with output files to serve that purpose.

The direct combat approach will be explained by giving an illustration of how the proposed discrete event combat model goes about the execution of a combat engagement.

Figure 12 shows two 5-3-4 combat networks, BLUE and RED. Notice that although both Forces have a 5-3-4 configuration, the unique combinations used are different. Specifically, in the BLUE Force, the way Sensors are distributed across Deciders is 3-1-1, while Influencers follow a 2-1-1 format. In the RED Force, the Sensors have a 2-2-1 format, while the Influencers have a 1-1-2 format.

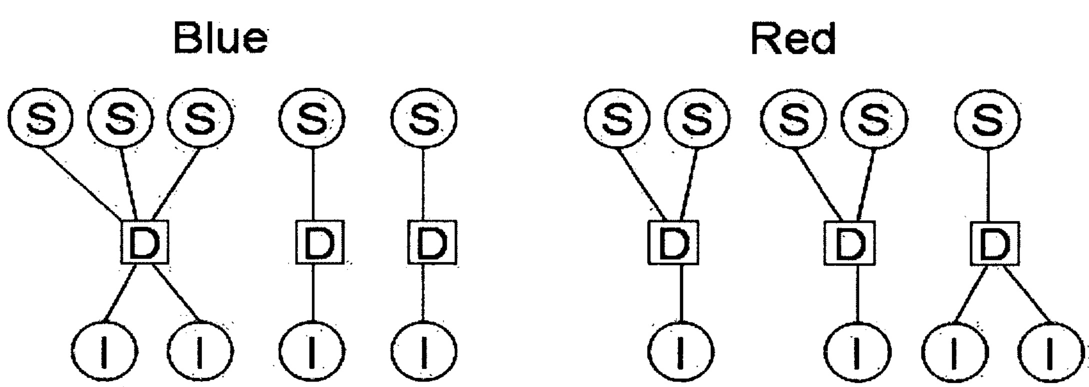

Figure 12. BLUE and RED 5-3-4 Combat Networks

The manner in which the proposed Combat Model decides who takes the first shot is by random number generation; a coin toss. Specifically, at each round or iteration, the number of Sensors is counted and a random number is generated based on the Sensors

count. For instance, the total Sensors count, BLUE and RED, in Figure 12 is 10. Therefore, the model generates a random number between 1 and 10. Say the generated number was 4. Since the 4th Sensor is in the BLUE Force, the BLUE Force takes the shot. Once the model determines who takes the first shot, another random number generation takes place to decide which adversary Sensor or Influencer is destroyed. Since there are a total of 5 Sensors and 4 Influencers (a total of 9 potential targets), the model generates a number between 1 and 9 and the selected target is then eliminated from that combat Force.

As we go through the iterations in this battle, it will be noticed that the model will only select a Sensor that is a part of a Combat Cycle. Specifically, the coin toss will never pick a Sensor that has no corresponding Influencer(s) that can take the shot and destroy an adversary Sensor or Influencer. This is an important function of the model and its importance will grow more as forces get depleted. This "one shot, one kill" approach is also important because as we iterate, the coin toss acts like a weighted average, where an advantage is given to the Force that has more Sensors. The logic behind this is that the Force with more Sensors is more likely to find a target.

We will now do the first iteration for the example in Figure 12. We will assume the first random number generated is a 7. Since the 7th Sensor is in the RED Force, the RED Force will take the first shot. The next step is to generate another random number which will determine the adversary target. Since there are a total of 9 potential targets, a random number is generated between 1 and 9. We will assume that number is 3. This means that the third Sensor which is connected to the first Decider in the BLUE Force is the acquired target. The RED Sensor transmits this information to the Decider to which it

is connected. The RED Decider instructs the Influencer attached to it to attack the acquired BLUE target. At that point, the BLUE target is destroyed. Figure 13 shows this engagement and the highlights RED Combat Cycle which destroyed the BLUE target.

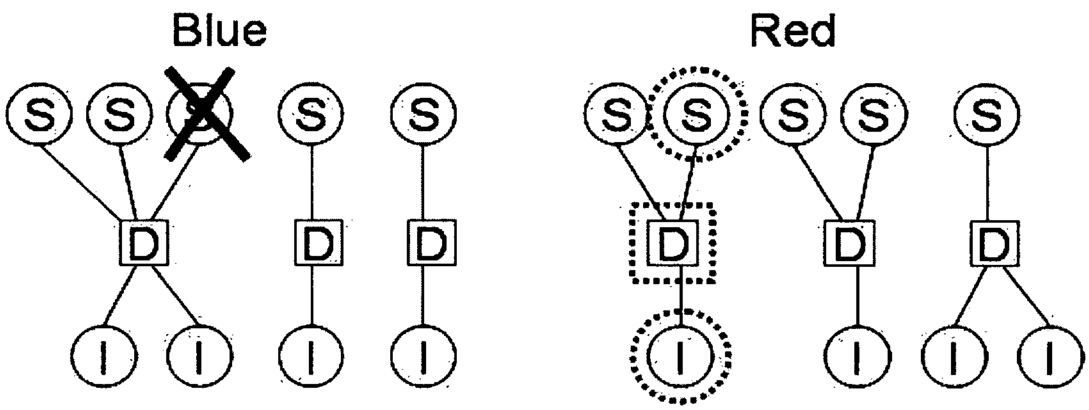

Figure 13. First Combat Cycle Iteration – RED Attacks BLUE

Another random number is generated to decide who takes the next shot. Since there are 9 remaining Sensors, that random number is going to be between 1 and 9. You can see that since the RED Force has more Sensors than the BLUE Force, it is more likely to acquire the next target. We will assume that number is 5. The RED Force takes the shot. A random number is generated to decide which target is eliminated. Say that number is 6. The target acquired is therefore the second Influencer connected to the first Decider of the BLUE Force. The RED Sensor sends this information to its Decider, and the Decider instructs its Influencer to eliminate the BLUE target. This RED Combat Cycle and the combat engagement are shown in Figure 14. A grey-colored sign is used to show the latest destroyed target.

Another random number is generated; let us say it is 1. The BLUE Force now takes a shot for the first time. A random number between 1 and 9 is generated to decide which RED target is destroyed, let us say it is 8. The RED target acquired is the first Influencer connected to the third Decider. The BLUE Sensor informs its Decider of the acquired target, and the Decider orders its remaining Influencer to attack and destroy the RED target. Figure 15 depicts this engagement and shows the BLUE Combat Cycle that acquired and destroyed the RED target.

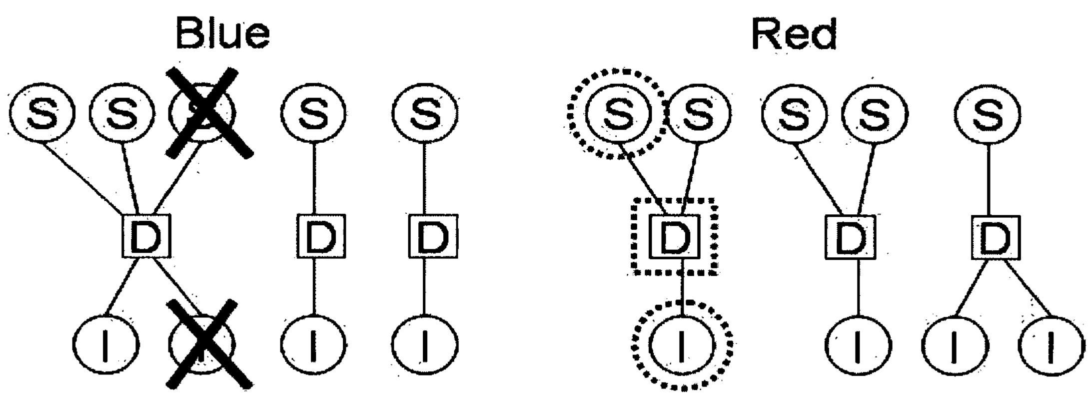

Figure 14. Second Combat Cycle Iteration – RED Attacks BLUE

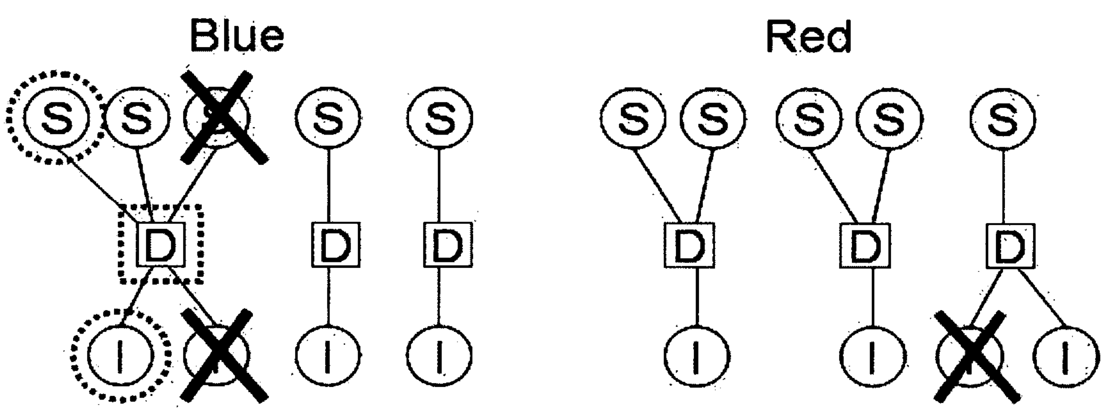

Figure 15. Third Combat Cycle Iteration – BLUE Attacks RED

Another round is initiated now and we will assume the random number is 9. The RED Force takes the next shot which means detection occurs through its fifth Sensor. A random number generation takes place to decide which BLUE target is acquired. Let us assume it yields a 5. This means that the remaining Influencer attached to the first BLUE Decider is the next target. Accordingly, the RED Sensor informs its Decider of the acquired target, and the Decider orders its Influencer to take it out. Figure 16 depicts this engagement. An important observation in Figure 16 is how the first and second BLUE Sensors are no longer a part of an effective Combat Cycle and have an NCC (No Combat Cycle) label on top of them. The same applies to their Decider. This is all due to the fact that their Decider has lost its firepower when its two Influencers were eliminated. Therefore, those two RED Sensors and their corresponding Decider are rendered combat ineffective.

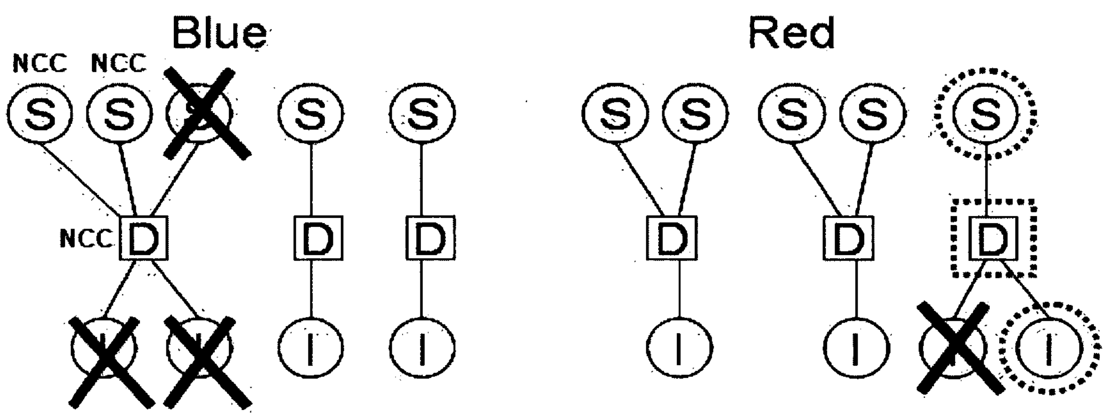

Figure 16. Fourth Combat Cycle Iteration – RED Attacks BLUE

Another engagement now takes place. Since there are only seven operational Sensors left in Battle, the random number generated will be between 1 and 7. Let the

random number this time be 6. This means the fourth RED Sensor gets selected and therefore the RED Force takes the next shot. Another random number is generated and this time it will be between 1 and 6 since there are six BLUE targets left. Let that number be 4. The fourth remaining BLUE Sensor is acquired as a target. Consequently, the RED Sensor sends the information to its Decider who will give orders to the sole Influencer it has to take that target out. Figure 17 shows this engagement. Notice that this engagement results in taking the second BLUE Decider and its Influencer out of battle as they are no longer part of a Combat Cycle.

The model keeps running as both Forces still have Combat Cycles left in their network. Notice that the RED Force is more likely to take the next shot because it possesses five Sensors compared to one Sensor the BLUE Force still have operational.

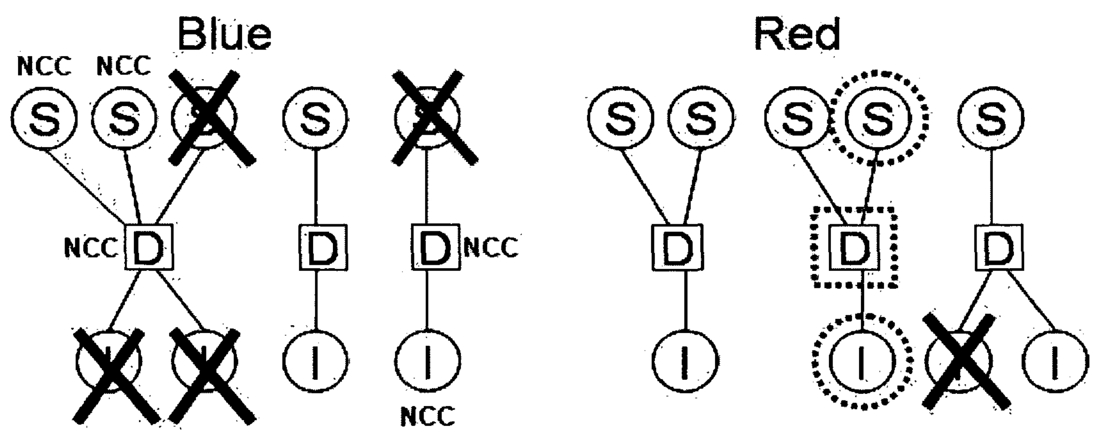

Figure 17. Fifth Combat Cycle Iteration – RED Attacks BLUE

A random number between 1 and 6 is generated and let us say it is 2. This means the first RED Sensor is selected and therefore the RED Force takes the next shot. Another random number between 1 and 5 is generated to decide the acquired target and let that

number be 3. This means the third BLUE Sensor (in order of targets remaining) is acquired as a target. The RED Sensor communicates this information to its Decider, and the Decider sends the Influencer to take out the BLUE target. This engagement effectively ends the battle as the BLUE Force is no longer combat effective as this engagement eliminated its only remaining Combat Cycle. The RED Force is victorious. Figure 18 illustrates the final engagement.

We can notice that the logic behind the Combat Model is not that difficult to understand. We now have an appreciation of the importance of Combat Cycles. Take the hypothetical case of the BLUE Force losing its first two Influencers at the beginning of battle. This will render their Decider, and the three Sensors connected to it, combat ineffective. They lost their firepower and can no longer form a functional Combat Cycle.

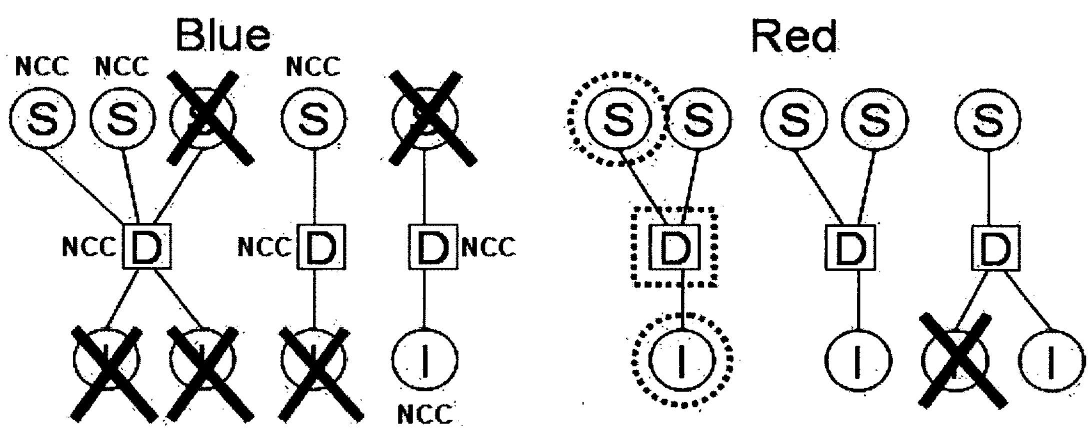

Figure 18. Final Combat Cycle Iteration – RED Attacks BLUE

It is clear how Combat Cycles play an important role in keeping assets operational on the battlefield. This issue will be examined in depth in this research as we start looking at the performance metrics of each combat configuration in battle. Those

performance metrics will enable us to quantitatively assess the strengths and weaknesses of each configuration.

The code for the Visual basic-based discrete event Combat Model is provided in Appendix F.

Now that this example is presented, we can understand the benefits of discrete event simulation in comparison with the agent-based approach Deller and Fidanci followed. The burden of non-value added activities or points of time in which no changes in the state of the system takes place is no longer there for the computer processors' to bear. Only those points in time in which a change in the state of the system takes place are represented. A more formal explanation of the speed-up and efficiency of the proposed discrete event combat model will be given in Section 3.8.

3.6 The Conceptual Model: A High-Level View

After illustrating the direct combat concept this research followed in the development of the combat model, now would be an appropriate time to give a high-level view of the model. Figure 19 presents this flow-chart-like view.

The input setup starts with a coin-toss for sensing using the weighted average as explained in the previous section. Depending on the generated random number, the simulation gives the first shot to the RED or BLUE force. Then another coin toss is carried out to determine which target is acquired. When the target is acquired, the Decider instructs an influencer to eliminate that target. After the elimination of the target, the simulation checks if there are any remaining Combat Cycles on both sides, if one side

does not have a functional Combat Cycle left, the simulation ends. Otherwise, it will go to the next iteration.

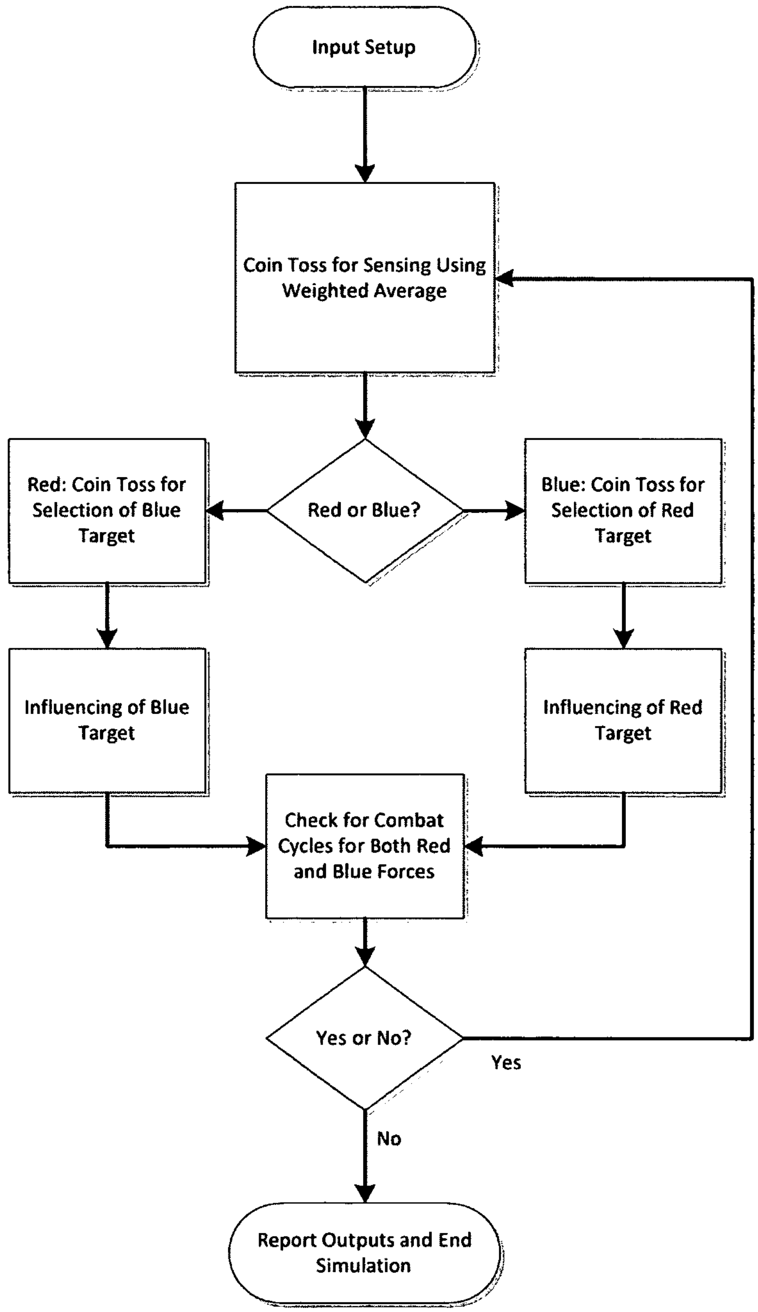

Figure 19. High Level View of Combat Model

3.7 Segregation of Simulation Runs

As the combat model was developed, it was decided to segregate the simulation runs. This was not a requirement for programming considerations, as the model could be set up to run all unique engagements (RED vs. BLUE combat) and output the data with a single click of a mouse. Rather, this decision was taken for the following considerations:

1. This will significantly reduce the total simulation running time as will be explained shortly. Therefore, it will help expedite the data collection and generation process by enabling simultaneous running of different configurations. Segregating the runs basically facilitates the use of different computers (i.e. processors) at the same time.
2. Having different machines enables the collection of preliminary data that can be used to verify the Combat Model and compare it with the results of Deller (2009) and Fidanci (2010).
3. This enables the early detection of errors and consequent debugging of the combat model.

Accordingly, Excel was used to help in segregating the simulation runs. Essentially, it was a simple arbitrary process, where a reasonable amount of unique engagements was assigned to each instance (i.e. computer) running the combat model. Table 9 illustrates how the simulation runs were segregated. If you sum up the contents of the last column in Table 9, the result would be 3,724,225. If you recall, this is the total number of unique engagements conducted for this research. The way this exactly works is that for each instance of the Combat Model, a lower and upper bound (Low and High

in Table 9) is given to it. Those bounds are represented as variables in the Combat Model, “lowerbound” and “upperbound”, and are altered according to the instance that needs to be run.

Table 9. Lower and Upper Bounds for Different Instances of the Combat Model

Computer	Low	High	Unique Engagements
1	1	105	750,531
2	106	137	837,676
3	138	145	937,013
4	146	193	603,623
5	194	285	595,382

The full potential of this segregation concept would be realized at any future research attempts that can improve the speed and efficiency of the combat model. As for the estimated running time this would take, a test run was conducted to reach this estimate. Specifically, the code for the combat model had to be adjusted to record the time it takes to complete a certain number of simulations. Therefore, it was decided to conduct a test run, have it run for a couple of hours while keeping track of time, and output that time to the same output file that gives the percentage of win for a certain combination over another. Configuration 194, or 11-5-11, was selected for this purpose. The rationale behind selecting this particular configuration is that 194 is one of the largest configurations ran in this research and it would therefore give a conservative estimate. Thus, it was run for a couple of hours and stopped. When the simulation was stopped, there was a total of 34957 replication executed.

The time recorded was 2209.796 seconds. This was averaged and it gave an estimated 0.063215 second per engagement. Since the number of engagements to be run

in this research is 3,724,225 (see Table 6), the estimated time per engagement was multiplied by this number and then divided by 86,400 (number of seconds per day). The result was almost 2.7 days of running time. However, by utilizing 5 machines and breaking up the engagements as explained previously, this number would go down to 0.54 days or around 13 hours.

It is clear from the aforementioned time calculations that the proposed discrete event combat model exhibits significantly higher levels of efficiency and speed compared to the previous agent-based modeling approaches adopted by Deller and Fidanci. Running the unique combinations of one configuration took Deller's model 78 hours to complete. Fidanci's 55 configurations required three months to finish on several machines. Compare that to the time it took this research to run 285 configurations and you will see that the time savings and efficiency jumps are indeed very significant.

The code for the adjusted Combat Model for time-keeping purposes is shown in Appendix M. The specifications for the computers used to run the experiments are given in Appendix N. Those specifications were obtained by using a program called "Belarc Advisor", which can build a detailed a profile for a computer. This program can be found here: http://www.belarc.com/free_download.html.

3.8 Model Verification Overview

Previous attempts at modeling the information age combat model by Deller (2009) and Fidanci (2010) used agent-based modeling. This research utilizes the discrete event simulation paradigm for its perceived increases in efficiency. This research, reaping the benefits of this jump in simulation speed and efficiency, will run 285 configurations.

Specifically, this research will focus on unbalanced force sizes (X-Y-Z) doing symmetric battle (X-Y-Z force against an identical X-Y-Z force). Both Deller (2010) and Fidanci (2011) conducted their experiments with symmetric force sizes (X-Y-X) doing symmetric battle as well, with the difference being that Fidanci conducted more experiments than Deller. The work of Deller (2009) was chosen as a baseline for comparison. Therefore, upon the completion of the simulation runs in this research, the data will be compared with the data from his work for those experiments that are in common. This is how theoretical results will be used to verify the proposed Combat Model. Model verification results will be discussed in Section 4.1.

In simulation modeling and analysis, normally there are two steps associated with the development of models, which are verification and validation. While verification will be addressed in this research, validation cannot be carried out due to the lack of real world data. Validation is defined is an iterative process that compares the model behavior to the actual system behavior, and use the inconsistencies that result from this comparison to improve the model. However, since there are real world data available, validation of the model developed in this research was not done.

3.9 Performance Metrics

Visual Basic programs were written to calculate the performance metrics from the output files. Focus will be on the following metrics (Deller, 2009; Fidanci, 2010):

1. ***Disparity***: disparity is the sum of the inequality of the way Sensors and Influencers are distributed across the Deciders in the network. Disparity can be expressed in mathematical terms as:

Disparity = [max(S*n*)-min(S*n*)] + [max(I*n*)-min(I*n*)]

where, S*n* = the number of Sensors assigned to each of *n* Deciders

I*n* = the number of Influencers assigned to each of *n* Deciders

The higher the disparity value, the higher the likelihood is to get an extremely high or low value for the probability to win. The way in which disparity affects the probability of win is determined by the manner in which Sensors and Influencers are distributed across Deciders; if there is a degree of balance. For instance, if a Decider has 4 or 5 Influencers and only 1 Sensor, then it would only take the destruction of that one Sensor to make all Influencers combat ineffective. The Disparity Visual Basic program is shown in Appendix G.

2. ***Robustness***: robustness is defined as the minimum number of nodes lost that would render a particular combat configuration unable to produce a combat cycle, and therefore, unable to engage enemy targets. Mathematically, robustness can be expressed as:

$\text{Robustness} = [\min(S_1, I_1)] + [\min(S_2, I_2)] + \cdots + [\min(S_n, I_n)]$

Where, S_n = the number of Sensors connected to Decider *n*.

I_n = the number of Influencers connected to Decider *n*.

The higher the robustness value, the higher the probability of a win. The Robustness Visual Basic program is shown in Appendix H.

3. ***Strength***: Strength is the sum of weighted average according to the logarithmic function of each decider, and reflects how many nodes of Sensors and Influencers linked to each Decider so that the combination stays combat effective. It is essentially the strength of connectivity. It can be expressed as:

$$\text{Strength} = \sum_{i=1}^{n} [\log_{10}(\#\text{ of Sensor}_i + 1) * \log_{10}(\#\text{ of Influencer}_i + 1)]$$

As its name indicates, the higher the value of the Strength of a combination, the higher the probability of a win is. The Disparity Visual Basic program is presented in Appendix I.

4. ***Power***: Power is another sum of weighted average, but according to the square root function of each decider. It shows how many nodes of Sensors and Influencers linked to each decider so that the combination stays combat effective. It can be formulated as:

$$\text{Power} = \sum_{i=1}^{n} [(\text{Sqrt}(\#\text{ of Sensor}_i)) * (\text{Sqrt}(\#\text{ of Influencer}_i))]$$

The higher the power value, the higher the probability of a win. The Power Visual Basic program is shown in Appendix J.

5. *Stability*: refers to the stability of Deciders. It is the sum of the quotient of the number of Sensors and Influencers connected to each Decider. This metric has a negative correlation with combat effectiveness, and is an indication of Deciders being used in an ineffective manner. It is expressed mathematically as:

$$\text{Stability} = \sum_{i=1}^{n} [\text{Quotient } (\# \text{ of Sensor}_i, \# \text{ of Influencer}_i)]$$

The Stability Visual Basic program is shown in Appendix K.

6. *Connectivity*: The connectivity metric is the sum of the absolute differences between the number of Sensors and Influencers connected to each Decider. This metric has a certain degree of negative correlation with combat effectiveness. It is expressed mathematically as:

$$\text{Connectivity} = \sum_{i=1}^{n} [\text{ABS } (\# \text{ of Sensor}_i - \# \text{ of Influencer}_{i)}]$$

The Connectivity Visual Basic program is shown in Appendix L.

3.10 Data Analysis

Performance metrics which will be utilized in this study were discussed in the previous section. The utility of those metrics is very significant to the task of gaining insights into the impact of arranging a force for networked effects on the force's combat effectiveness. For instance, does a battle configuration that possesses a more balanced distribution of sensing and influencing capabilities across commanding or deciding units have any advantage in combat over a force with equal assets and capabilities but with less

balance in the arrangement of its assets? The calculation of those metrics will help in addressing this question for the case of unbalanced forces fighting symmetric wars, and many other questions.

Regression analysis will be used to provide basis for predicting the probability of win for a configuration. This research will conduct the following regression analyses:

1. Multiple regression for all of the metrics combined.
2. Multiple nonlinear regression for all of the metrics combined.

This research, unlike the previous endeavors in the same area, carries out symmetric engagements for *unbalanced* combat configurations. The regression analysis will help in determining the significance of the metrics as predictive factors in the case of unbalanced combat configurations doing symmetric warfare. This will help in revealing which metrics wield the highest predictive power over a configuration's probability of defeating an identical one. Moreover, this will also assist in making a comparison with the metrics that were found to be powerful predictors in the case of symmetric battle configurations doing symmetric warfare.

All this analysis will certainly provide valuable information about the utility of network centric warfare, especially in symmetric battle configurations. More importantly, it will help in achieving the stated purpose of this research which is bringing the dilemma of quantifying the impact of network effects on combat effectiveness closer to resolution. This due to the fact that it will present an opportunity to assess whether forces that are organized for better networked effects (i.e. forces that have favorable robustness,

strength, power, etc scores) have any advantage in better-leveraging its combat effectiveness over forces with arrangements of less networked effects.

CHAPTER 4

RESULTS AND DATA ANALYSIS

This chapter discusses the results and data analysis. It is organized into eight sections. It will start with explaining the manner in which overview of model verification. Then the results of the linear and non-linear regression analyses will be presented and discussed. Those analyses will be carried out on the aggregated data and then the data will be disaggregated to focus on high percentage win engagements and evenly-matched engagements. The focus will be on highlighting the utility of the performance metrics defined and used in previous research attempts that studied balanced configurations, in the case of unbalanced forces configurations doing symmetric engagements.

4.1 Model Verification

One of the most important steps in any modeling and simulation research is the verification of the model. Verification is done to ensure that the model is running in the way it was envisioned and implemented to run and that the algorithms incorporated in it are working properly. The way the verification of the model developed in this research was approached is by comparing the results with those of Deller (2009). Originally, the idea was to compare the regression analysis between the current model's results and the previous results of Deller (2009). However, it was quickly apparent that this was not a good approach to the verification because the models were developed using two different modeling paradigms. Thus, an alternative approach to the model verification was conceived using the ordinal ranking of the various configurations. Specifically, the

ordinal rankings of Deller's results were compared to the ordinal rankings of this research's results. The verification was done using the three configurations that were analyzed by Deller (2009) which are the 7-3-7, 8-3-8, and 9-5-9 configurations. Additionally, the ordinal rankings were compared using different numbers of replications in order to examine the impact of replications on the modeling results. The comparison was done by ordering the unique combinations in terms of their average win percentage. Then the average difference between the current model's ordinal rankings and Deller's ordinal rankings was calculated.

Table 10 shows a summary of the verification results for the 7-3-7 configuration. For instance, at 30 replications, the average difference in ordinal rank between the two models was 8.5. This means that on average the rankings differed by 8.5 ranks. As shown by Table 1, the average ordinal difference decreased as the number of replications was increased. Consequently, as the number of replications was increased the current model more closely reflected the results of Deller (2009). This shows that the current model is working in the way it was intended, modeling the logic of the Information Age Combat Model.

Table 10. Average Ordinal Difference for the 7-3-7 Configuration

Replications	Average Ordinal Difference
30	8.5
100	6.3
1000	3.1

Figure 20 depicts a graphical representation of the relationship between the number of replications and the average difference between the ordinal rankings of the results from the two different models, for the 7-3-7 configuration.

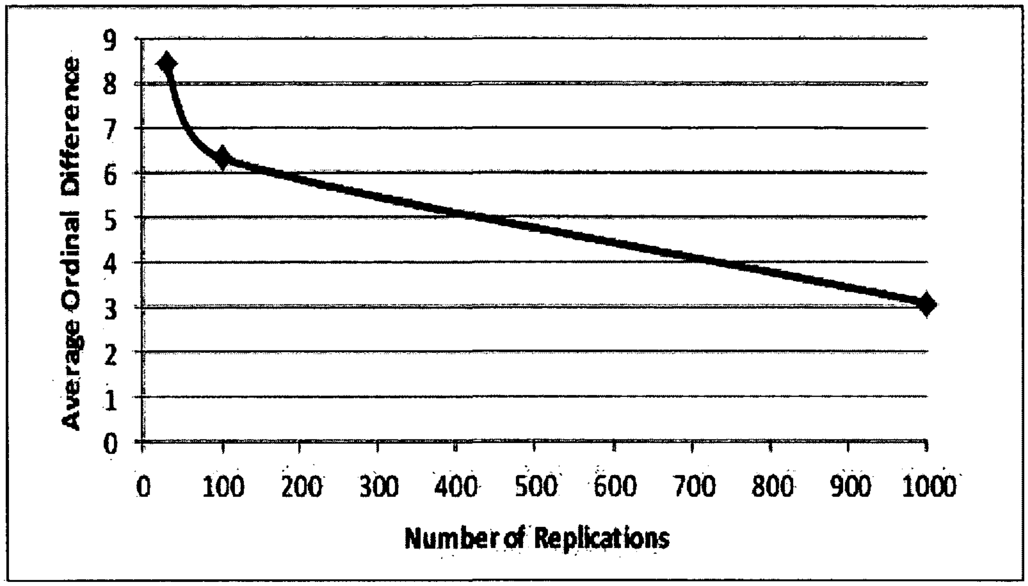

Figure 20. Average Ordinal Difference vs. Replications for the 7-3-7 Configuration

Table 11 shows a summary of the verification results for the 8-3-8 configuration. For instance, at 30 replications, the average difference in ordinal rank between the two models was 11.0. This means that on average the rankings differed by 11.0 ranks. As shown by Table 2, the average ordinal difference decreased as the number of replications was increased. Subsequently, as the number of replications was increased the current model more closely reflected the results of Deller (2009). Once again, this shows that the current model is working in the way it was intended, modeling the logic of the Information Age Combat Model.

Table 11. Average Ordinal Difference for the 8-3-8 Configuration

Replications	Average Ordinal Difference
30	11.0
100	7.6
1000	4.3

Figure 21 displays a graphical representation for the 8-3-8 configuration of the relationship between the number of replications and the average difference between the ordinal rankings of the results from the two different models.

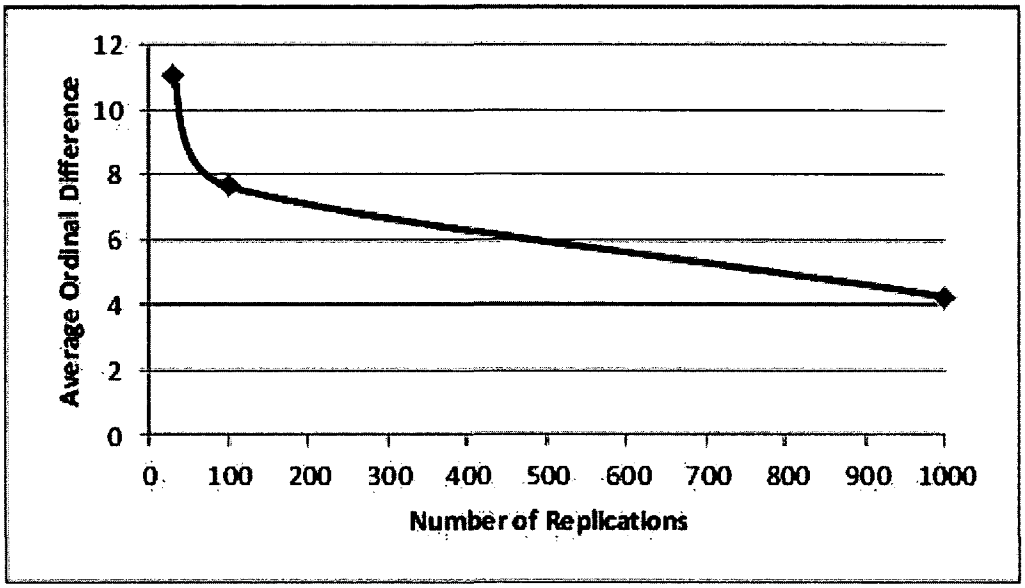

Figure 21. Average Ordinal Difference vs. Replications for the 8-3-8 Configuration

Table 12 presents a summary of the verification results for the 9-5-9 configuration. For example, at 30 replications, the average difference in ordinal rank between the two models was 17.7. This means that on average the rankings differed by 17.7 ranks. As displayed in Table 3, the average ordinal difference decreased as the number of replications was increased. Therefore, as the number of replications was increased the current model more closely reflected the results of Deller (2009). Once more, this verifies that the current model is working in the way it was intended, modeling the logic of the Information Age Combat Model.

Figure 22 graphically depicts the relationship between the number of replications and the average difference between the ordinal rankings of the results from the two different models, for the 9-5-9 configuration.

Table 12. Average Ordinal Difference for the 9-5-9 Configuration

Replications	Average Ordinal Difference
30	17.7
100	14.3
1000	10.0

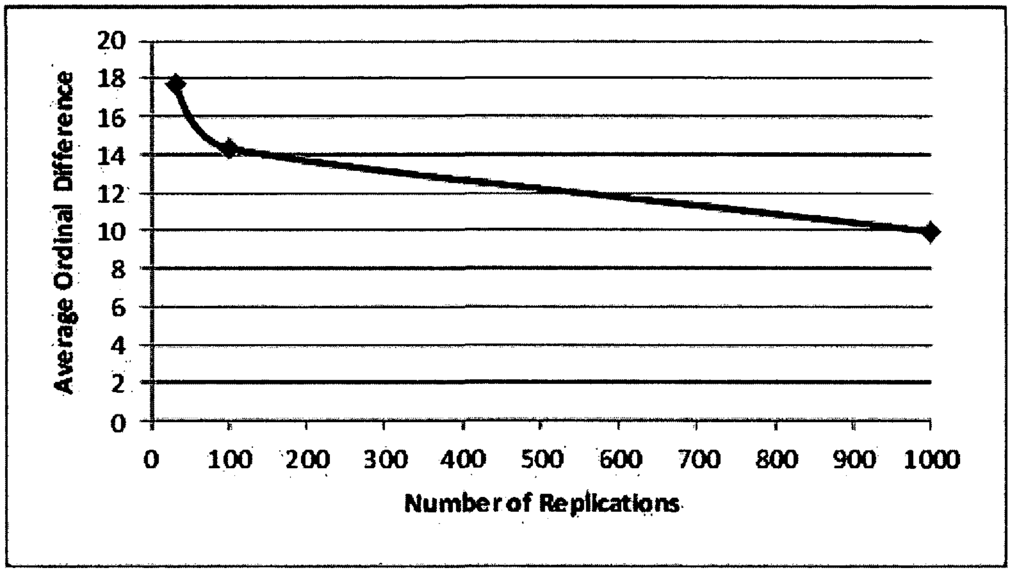

Figure 22. Average Ordinal Difference vs. Replications for the 9-5-9 Configuration

4.2 Linear Regression - Aggregated Data

JMP is the statistical software package used to conduct the analysis needed for this simulation. The main reason for choosing JMP was because of its ability to explore virtually unlimited amount of data, compared to other packages. A summary of the linear regression analysis for main effects using the aggregated data is shown in Figure 23 below.

The R-Squared value is 0.54. The Parameter estimates portion of the regression shows that all metrics are considered significant, except Rob_Red. Rob_Blue was zeroed out because it has a collinear relationship with Rob_Red. The reason why Rob_Blue was zeroed out, and not Rob_Red, is simply because of the order in which the variables were added to the regression. Since Rob_Red is insignificant, it will be removed and the regression will be run again to include only the significant variables. Figure 24 shows the results of the regression analysis including only the significant variables.

Summary of Fit

RSquare	0.548911
RSquare Adj	0.548909
Root Mean Square Error	1.654948
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	11	12412008	1128364	411984.3
Error	3.7e+6	10200071	2.738853	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Parameter Estimates

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept		50.016278	0.011075	4516.2	<.0001*
Conn_Red	Biased	0.1100074	0.00324	33.95	<.0001*
Conn_Blue	Biased	-0.117425	0.000964	-121.8	<.0001*
Disp_Red		0.0591381	0.000709	83.44	<.0001*
Disp_Blue		-0.057387	0.000709	-80.97	<.0001*
Pow_Red	Biased	3.1559992	0.01151	274.21	<.0001*
Pow_Blue	Biased	-3.155595	0.01151	-274.2	<.0001*
Rob_Red	Biased	-0.013155	0.007525	-1.75	0.0804
Rob_Blue	Zeroed	0	0	.	.
Stab_Red		0.1238072	0.000928	133.34	<.0001*
Stab_Blue		-0.120581	0.000928	-129.9	<.0001*
Stre_Red		-1.225003	0.075303	-16.27	<.0001*
Stre_Blue		1.3110649	0.075303	17.41	<.0001*

Figure 23. Linear Regression Results for Main Effects

The R-squared value is 0.54. All the metrics are significant. However, the issue of multicollinearity between the performance metrics must be examined more closely. For this purpose, this research will utilize the Variance Inflation Factor (VIF). When using VIFs, as a rule of thumb, values exceeding 10 are considered strongly collinear and must be removed one at a time from the regression model until the model stabilizes and all the VIFs take values of fewer than 10.

Figure 25 shows the VIF values for each of the significant metrics. The values show that there is a multicollinearity issue that must be addressed. Therefore, the metric with the highest VIF must be removed. Since two metrics have the highest VIF, which are Pow_Red and Pow_Blue, one of them must be removed and the regression must be rerun again. As explained before, this procedure will be followed until all VIFs have values of fewer than 10. Only then can a regression model be presented as not having serious multicollinearity concerns. Accordingly, this procedure was followed until a final regression model was obtained. The final regression model is presented in Figure 26.

Therefore, with the data aggregated, the Connectivity, Disparity, and Stability metrics, as well as the Red Power, give us the best predictive capability. Forty-seven percent of the variation in the results is explained by those metrics. It is noteworthy that this research decided to exclude eigenvalues, which is a metric utilized in previous research attempts, because its value as a distinguishing factor between configurations diminishes as the number of unique combinations gets larger. Therefore, although this exclusion is justified in terms of the overall purpose of this research, it is certain to have some impact, though not significant, on the regression results, especially for those configurations that have a low number of unique combinations.

Summary of Fit

RSquare	0.54891
RSquare Adj	0.548909
Root Mean Square Error	1.654948
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	12412000	1241200	4531822
Error	3.7e+6	10200079	2.738854	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	50.014713	0.011039	4530.9	<.0001*
Conn_Red	0.11552	0.000747	154.68	<.0001*
Conn_Blue	-0.11849	0.000747	-158.7	<.0001*
Disp_Red	0.0588785	0.000693	84.96	<.0001*
Disp_Blue	-0.057647	0.000693	-83.18	<.0001*
Pow_Red	3.1497463	0.01094	287.92	<.0001*
Pow_Blue	-3.161848	0.01094	-289.0	<.0001*
Stab_Red	0.1235946	0.00092	134.27	<.0001*
Stab_Blue	-0.120794	0.00092	-131.2	<.0001*
Stre_Red	-1.222972	0.075294	-16.24	<.0001*
Stre_Blue	1.3130963	0.075294	17.44	<.0001*

Figure 24. Linear Regression Analysis for Significant Main Effects

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	50.014713	0.011039	4530.9	<.0001*	
Conn_Red	0.11552	0.000747	154.68	<.0001*	6.4643982
Conn_Blue	-0.11849	0.000747	-158.7	<.0001*	6.4643982
Disp_Red	0.0588785	0.000693	84.96	<.0001*	3.0999302
Disp_Blue	-0.057647	0.000693	-83.18	<.0001*	3.0999302
Pow_Red	3.1497463	0.01094	287.92	<.0001*	149.8445
Pow_Blue	-3.161848	0.01094	-289.0	<.0001*	149.8445
Stab_Red	0.1235946	0.00092	134.27	<.0001*	3.7858526
Stab_Blue	-0.120794	0.00092	-131.2	<.0001*	3.7858526
Stre_Red	-1.222972	0.075294	-16.24	<.0001*	128.15259
Stre_Blue	1.3130963	0.075294	17.44	<.0001*	128.15259

Figure 25. VIF Values for Significant Main Effects

More importantly, it is likely there are other unknown metrics that play a role in determining the outcome of engagements in the case of unbalanced configurations doing symmetric engagements. The metrics previously defined have substantial impact on the outcome of balanced forces doing asymmetric engagements as shown in the literature. This research, on the other hand, represents the first effort of its kind that studies symmetric engagements of unbalanced configurations. And while the linear regression results show that most of the metrics were also significant in the case of unbalanced forces, it is obvious that there is a need for more research to be done to come up with and test other metrics as well. Complete linear regression results can be found in Appendix P.

Summary of Fit

RSquare	0.474835
RSquare Adj	0.474834
Root Mean Square Error	1.785668
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	7	10737006	1533858	481042.9
Error	3.7e+6	11875073	3.188609	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	48.940022	0.011803	4146.5	<.0001*	.
Conn_Red	-0.321399	0.000524	-612.8	<.0001*	2.7384091
Conn_Blue	0.3411684	0.000486	702.06	<.0001*	2.3508426
Disp_Red	-0.038432	0.000502	-76.58	<.0001*	1.3964392
Disp_Blue	0.0264916	0.000507	52.27	<.0001*	1.4241937
Pow_Red	0.115944	0.001142	101.54	<.0001*	1.4023581
Stab_Red	-0.171693	0.000835	-205.5	<.0001*	2.6788433
Stab_Blue	0.1598862	0.000844	189.39	<.0001*	2.7351894

Figure 26. Final Regression Model for All Metrics

4.3 Linear Regression - Disaggregated Data: High Percentage Win Engagements

The mean for the aggregated data was found to be 50%. Therefore, the data was disaggregated in an attempt to get more insights. Therefore, all the unique engagements in which the RED force had a percentage win of greater than or equals 55% AND less than or equals to 45% were selected for analysis. This basically includes around 185,000 engagements in which an advantage in combat was present and helped achieve victory. This also excludes around 3,500,000 engagements in which the battle was very close and evenly-matched. A summary of the linear regression results is shown in Figure 27.

Summary of Fit

RSquare	0.896033
RSquare Adj	0.896027
Root Mean Square Error	2.025086
Mean of Response	50.15859
Observations (or Sum Wgts)	184559

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	11	6522637.9	592967	144591.7
Error	184547	756822.6	4.100975	Prob > F
C. Total	184558	7279460.5		<.0001*

Parameter Estimates

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept		49.734743	0.058668	847.73	<.0001*
Conn_Red	Biased	-0.341207	0.016111	-21.18	<.0001*
Conn_Blue	Biased	0.1291031	0.005122	25.20	<.0001*
Disp_Red		0.2176863	0.003852	56.51	<.0001*
Disp_Blue		-0.220201	0.003837	-57.38	<.0001*
Pow_Red		0.156223	0.053094	2.94	0.0033*
Pow_Blue		0.3051155	0.053049	5.75	<.0001*
Rob_Red	Biased	-0.59275	0.038023	-15.59	<.0001*
Rob_Blue	Zeroed	0	0	.	.
Stab_Red		0.1561769	0.005267	29.65	<.0001*
Stab_Blue		-0.148304	0.005254	-28.22	<.0001*
Stre_Red		24.100095	0.372054	64.78	<.0001*
Stre_Blue		-22.41078	0.370091	-60.55	<.0001*

Figure 27. Linear Regression Results for High Percentage Win Engagements

The R-Squared value has significantly increased from 0.47 to 0.89. However, the VIF indicator must be taken into account to ensure that multicollinearity issues are addressed. After addressing the multicollinearity issue, the regression results shown in Figure 28 were obtained. After resolving the multicollinearity issues, the R-Squared value becomes 0.87. This shows that a force can be organized for better networked effects and consequently, better combat effectiveness, for some types of engagements. In those engagements, the Connectivity, Disparity, and Stability metrics, as well as the Red Power, played the most dominant role in determining the outcome of battle in favor of the RED force. However, this represents only a small part of the engagements. In the next section, the remaining engagements will be examined.

Summary of Fit

RSquare	0.879206
RSquare Adj	0.879201
Root Mean Square Error	2.1828
Mean of Response	50.15859
Observations (or Sum Wgts)	184559

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	7	6400145.5	914307	191895.0
Error	184551	879315.0	4.764618	Prob > F
C. Total	184558	7279460.5		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	48.492431	0.062619	774.40	<.0001*	.
Conn_Red	-0.574898	0.002852	-201.6	<.0001*	5.3295809
Conn_Blue	0.5814411	0.002725	213.36	<.0001*	4.7875482
Disp_Red	-0.140922	0.002554	-55.17	<.0001*	1.8443624
Disp_Blue	0.0927321	0.002593	35.77	<.0001*	1.8933631
Pow_Red	0.2629198	0.006208	42.35	<.0001*	2.3255871
Stab_Red	-0.332894	0.004536	-73.39	<.0001*	4.1523206
Stab_Blue	0.2826328	0.004601	61.42	<.0001*	4.2514644

Figure 28. Final Regression Results for High Percentage Win Engagements

4.4 Linear Regression - Disaggregated Data: Evenly-Matched Engagements

After excluding the engagements which had a somewhat high percentage win, linear regression analysis was carried out on the remaining engagements. Those remaining engagements represent around 3,500,000 engagements out of the almost 3,700,000 ran in this research. A summary of the regression results is shown in Figure 29. It is noticed that Rob_Red is insignificant. Rob_Blue was zeroed out because it has a collinear relationship with Rob_Red. After removing those metrics and also using the VIF indicator, the regression results shown in Figure 30 are obtained.

Summary of Fit

RSquare	0.446756
RSquare Adj	0.446754
Root Mean Square Error	1.547812
Mean of Response	49.99297
Observations (or Sum Wgts)	3539666

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	11	6847782	622526	259849.0
Error	3.5e+6	8480024	2.395721	Prob > F
C. Total	3.5e+6	15327806		<.0001*

Parameter Estimates

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept	Biased	50.023805	0.010653	4695.6	<.0001*
Conn_Red	Biased	0.041789	0.003229	12.94	<.0001*
Conn_Blue	Biased	-0.047967	0.000951	-50.42	<.0001*
Disp_Red		0.0626582	0.000682	91.84	<.0001*
Disp_Blue		-0.061361	0.000682	-89.92	<.0001*
Pow_Red	Biased	2.3854515	0.011344	210.29	<.0001*
Pow_Blue	Biased	-2.387971	0.011344	-210.5	<.0001*
Rob_Red	Biased	-0.008375	0.007501	-1.12	0.2642
Rob_Blue	Zeroed	0	0	.	.
Stab_Red		0.1116972	0.00089	125.55	<.0001*
Stab_Blue		-0.108872	0.00089	-122.4	<.0001*
Stre_Red		-0.470319	0.07297	-6.45	<.0001*
Stre_Blue		0.5338632	0.072979	7.32	<.0001*

Figure 29. Linear Regression Results for Evenly-Matched Engagements

Figure 30 shows that the Connectivity, Disparity, and Stability metrics, as well as Red Power, are the most significant variables. However, the R-Squared value is 0.39. Therefore, again we can conclude that the current metrics do not tell the whole story in the case of unbalanced force configurations doing symmetric engagements, especially for those engagements that were very evenly-matched and in which no decisive advantage was established in victory. This is evident in that fact that almost 61% of the variation remains unexplained by the current metrics. More research needs to be to uncover other potentially significant metrics. The results for the aggregated data and the disaggregated data for evenly-matched metrics also lead to an important insight which will be discussed in Section 4.8.

Summary of Fit

RSquare	0.390428
RSquare Adj	0.390427
Root Mean Square Error	1.624695
Mean of Response	49.99297
Observations (or Sum Wgts)	3539666

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	7	5984407	854915	323876.5
Error	3.5e+6	9343400	2.639634	Prob > F
C. Total	3.5e+6	15327806		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	49.264978	0.011053	4457.4	<.0001*	.
Conn_Red	-0.291154	0.000491	-593.3	<.0001*	2.5642329
Conn_Blue	0.3037703	0.000455	668.01	<.0001*	2.2031294
Disp_Red	-0.007669	0.000475	-16.14	<.0001*	1.3578839
Disp_Blue	-0.001921	0.00048	-4.01	<.0001*	1.3843299
Pow_Red	0.0811687	0.00108	75.18	<.0001*	1.3584615
Stab_Red	-0.09999	0.000785	-127.4	<.0001*	2.6185727
Stab_Blue	0.0917402	0.000793	115.69	<.0001*	2.6720045

Figure 30. Final Regression Results for Evenly-Matched Engagements

4.5 Non Linear Regression - Aggregated Data

Non-linear regression is an analysis that attempts to find a nonlinear model of the relationship between the dependent variable and a set of independent variables. The non-linear regression analysis this research will use will cover two-way interactions and quadratic terms. Figure 31 presents a summary of the regression results. Table 16 in Appendix O gives the parameter estimates for each of the terms as well as their significance. We notice that nonlinear regression for the aggregated data gave a slightly higher R-Squared value. It is now 0.57, which shows that the main and interaction effects wield slightly stronger predictive power over the outcome of battle compared to linear regression.

Summary of Fit

RSquare	0.576261
RSquare Adj	0.576252
Root Mean Square Error	1.604007
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	81	13030464	160870	62526.26
Error	3.7e+6	9581615	2.572838	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Figure 31. Non Linear Regression Results - Aggregated Data

Table 16 shows that all the main effects are significant. Several interactions were found to be insignificant. The Rob_Blue main effect and several interaction effects were automatically zeroed out by JMP because of collinear relationships. The regression analysis was rerun again to get a cleaner model after removing insignificant terms and

those that were zeroed out. The results of the regression rerun are shown in Figure 32. Table 17 in Appendix O gives the parameter estimates for each of the terms and evaluates their significance. Table 17 shows that there are still some insignificant interaction terms. Therefore, the regression was rerun another time to reach a cleaner model that only has the significant terms. The results are shown in Figure 33. Table 18 in Appendix O gives the parameter estimates for each of the terms and evaluates their significance.

Summary of Fit	
RSquare	0.575861
RSquare Adj	0.575855
Root Mean Square Error	1.604758
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	50	13021407	260428	101127.4
Error	3.7e+6	9590671	2.575248	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Figure 32. Non Linear Regression Second Iteration - Aggregated Data

Summary of Fit	
RSquare	0.57586
RSquare Adj	0.575855
Root Mean Square Error	1.604758
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	46	13021395	283074	109921.0
Error	3.7e+6	9590684	2.575249	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Figure 33. Non Linear Regression Results Third Iteration - Aggregated Data

Now that all the insignificant terms were removed, the collinearity issue can be addressed further by utilizing the VIF. Figure 34 gives us a summary of the final regression results after taking the VIF values into account.

Summary of Fit

RSquare	0.525745
RSquare Adj	0.525743
Root Mean Square Error	1.696912
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	21	11888195	566105	196597.5
Error	3.7e+6	10723883	2.879511	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	48.460058	0.012732	3806.1	<.0001*	.
Conn_Red	-0.329117	0.000545	-604.1	<.0001*	3.2719051
Conn_Blue	0.3161851	0.000485	651.78	<.0001*	2.5941552
Disp_Red	-0.020879	0.000537	-38.86	<.0001*	1.7720196
Disp_Blue	0.075088	0.000506	148.45	<.0001*	1.5708418
Pow_Red	0.102599	0.001189	86.29	<.0001*	1.683821
Stab_Red	-0.013165	0.000889	-14.80	<.0001*	3.3623136
Stab_Blue	0.0651283	0.000857	75.95	<.0001*	3.1248402
(Conn_Red-6.49002)*(Disp_Red-6.912)	-0.012454	0.000191	-65.06	<.0001*	3.1255537
(Conn_Red-6.49002)*(Disp_Blue-6.912)	-0.010952	0.000174	-63.08	<.0001*	1.7271103
(Conn_Red-6.49002)*(Stab_Red-5.87047)	-0.024466	0.000293	-83.59	<.0001*	3.63413
(Conn_Red-6.49002)*(Stab_Blue-5.87047)	0.008551	0.00022	38.79	<.0001*	1.7624814
(Conn_Blue-6.49002)*(Disp_Red-6.912)	-0.014046	0.000151	-93.06	<.0001*	1.3048428
(Conn_Blue-6.49002)*(Disp_Blue-6.912)	0.0401561	0.000169	237.73	<.0001*	2.4339812
(Conn_Blue-6.49002)*(Stab_Red-5.87047)	-0.006308	0.000219	-28.83	<.0001*	1.735441
(Disp_Red-6.912)*(Pow_Red-9.3553)	0.1155271	0.000523	220.98	<.0001*	2.0712685
(Disp_Red-6.912)*(Stab_Red-5.87047)	0.0072216	0.000318	22.74	<.0001*	2.6598811
(Disp_Blue-6.912)*(Pow_Red-9.3553)	-0.090738	0.000523	-173.5	<.0001*	1.6813648
(Disp_Blue-6.912)*(Stab_Blue-5.87047)	0.0129346	0.000298	43.45	<.0001*	2.3374503
(Pow_Red-9.3553)*(Stab_Red-5.87047)	0.0879857	0.000583	150.92	<.0001*	1.6167937
(Stab_Red-5.87047)*(Stab_Red-5.87047)	-0.013524	0.000365	-37.01	<.0001*	2.5660761
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)	0.0393858	0.000287	137.17	<.0001*	1.5843972

Figure 34. Final Non Linear Regression Model - Aggregated Data

The final R-Squared value is 0.52. It is noticed that non-linear regression for the aggregated data has settled on the same main effects which linear regression gave. The percentage of variation explained by the main effects and the significant interaction effects is now 52% compared to the 47% which linear regression gave.

4.6 Non Linear Regression - Disaggregated Data: The High Percentage Win Engagements

The next step is carry out non linear regression for all the unique engagements in which the RED force had a percentage win of greater than or equals 55% and less than or equals to 45%. After removing the insignificant terms and resolving the collinearity issues using the VIF in the same manner as illustrated before, the final regression model obtained is shown in Figure 35.

The R-Squared is 0.90. Therefore, 90% of variation in the percentage win for all the unique engagements in which the RED force had a percentage win of greater than or equals 55% and less than or equals to 45% is explained by the included main and interaction effects. The R-Squared value for the same part of the data was 0.87. This again shows that a force can be organized for better networked effects and consequently, better combat effectiveness, for some types of engagements.

4.7 Non Linear Regression - Disaggregated Data: Evenly-Matched Engagements

The final step is to carry out non linear regression for those engagements that had a percentage win between 46% and 54%, which represent the majority of engagements. After removing the insignificant terms and resolving the collinearity issues using the VIF, the final regression model is shown in Figure 36.

Summary of Fit	
RSquare	0.901834
RSquare Adj	0.901821
Root Mean Square Error	1.967849
Mean of Response	50.15859
Observations (or Sum Wgts)	184559

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	24	65648652	273536	70636.77
Error	184534	714595.3	3.872431	Prob > F
C. Total	184558	72794605		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	51.474372	0.073305	702.20	<.0001*	.
Conn_Red	-0.789897	0.003023	-261.3	<.0001*	7.3670555
Conn_Blue	0.7576435	0.003212	235.85	<.0001*	8.1855585
Disp_Red	-0.041758	0.002734	-15.27	<.0001*	2.6002035
Disp_Blue	-0.03612	0.0029	-12.46	<.0001*	2.9144056
Pow_Blue	-0.016754	0.006855	-2.44	0.0145*	3.4398621
Stab_Red	-0.085826	0.004771	-17.99	<.0001*	5.6527232
Stab_Blue	0.0287758	0.004692	6.13	<.0001*	5.4383637
(Conn_Red-8.11646)*(Conn_Blue-8.16885)	0.0084268	0.001049	8.03	<.0001*	7.9128739
(Conn_Red-8.11646)*(Disp_Red-8.43161)	-0.028955	0.000805	-35.99	<.0001*	4.1617278
(Conn_Red-8.11646)*(Disp_Blue-8.43103)	0.0753669	0.000782	96.35	<.0001*	2.917949
(Conn_Red-8.11646)*(Pow_Blue-8.76703)	0.0470881	0.002356	19.99	<.0001*	4.7802699
(Conn_Red-8.11646)*(Stab_Red-6.43349)	-0.040148	0.001009	-39.78	<.0001*	2.8552621
(Conn_Red-8.11646)*(Stab_Blue-6.45094)	0.0440414	0.001083	40.65	<.0001*	3.274889
(Conn_Blue-8.16885)*(Disp_Red-8.43161)	-0.047145	0.000915	-51.53	<.0001*	3.8783109
(Conn_Blue-8.16885)*(Disp_Blue-8.43103)	-0.005288	0.000928	-5.70	<.0001*	5.5022449
(Conn_Blue-8.16885)*(Pow_Blue-8.76703)	-0.020944	0.002769	-7.56	<.0001*	6.523057
(Conn_Blue-8.16885)*(Stab_Red-6.43349)	-0.017499	0.001075	-16.28	<.0001*	3.1537558
(Conn_Blue-8.16885)*(Stab_Blue-6.45094)	0.0157057	0.001039	15.11	<.0001*	3.0020733
(Disp_Red-8.43161)*(Pow_Blue-8.76703)	0.0541937	0.002981	18.18	<.0001*	4.4954341
(Disp_Blue-8.43103)*(Pow_Blue-8.76703)	-0.138467	0.002813	-49.22	<.0001*	4.1159338
(Pow_Blue-8.76703)*(Stab_Blue-6.45094)	-0.131621	0.003503	-37.58	<.0001*	3.2681298
(Disp_Red-8.43161)*(Disp_Red-8.43161)	0.0023468	0.00078	3.01	0.0026*	2.4019222
(Disp_Blue-8.43103)*(Disp_Blue-8.43103)	-0.004877	0.000756	-6.45	<.0001*	2.2500637
(Pow_Blue-8.76703)*(Pow_Blue-8.76703)	0.086833	0.004961	17.50	<.0001*	3.2652274

Figure 35. Final Non Linear Regression Results for High Percentage Win Engagements

The final R-Squared value is 0.42. This is slightly higher than the final R-Squared the linear regression gave (0.39), which means a marginal increase in the predictive power when the interaction effects are accounted for. However, 42% of variation explained show that there is a need to do more research to uncover other potential metrics

that have a significant impact in the case of unbalanced force configurations doing symmetric engagements, as the majority of the engagements indicate evenly-matched configurations due to the nature of having configurations of the same size battling each other. The currently-defined metrics do not tell the whole story.

Summary of Fit

RSquare	0.425206
RSquare Adj	0.425202
Root Mean Square Error	1.577671
Mean of Response	49.99297
Observations (or Sum Wgts)	3539666

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	21	6517468	310356	124688.6
Error	3.5e+6	8810338	2.489046	Prob > F
C. Total	3.5e+6	15327806		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	49.825551	0.013656	3648.5	<.0001*	.
Conn_Red	-0.32698	0.000506	-646.2	<.0001*	2.8914467
Conn_Blue	0.3206684	0.000547	586.62	<.0001*	3.3762676
Disp_Red	-0.004091	0.000526	-7.77	<.0001*	1.7685084
Disp_Blue	0.0092326	0.000522	17.69	<.0001*	1.7396659
Pow_Blue	0.0085862	0.001279	6.72	<.0001*	2.0217339
Stab_Red	0.0153997	0.000861	17.88	<.0001*	3.3405512
Stab_Blue	-0.010352	0.000856	-12.10	<.0001*	3.3001059
(Conn_Red-6.40522)*(Disp_Red-6.83277)	-0.030625	0.000184	-166.8	<.0001*	2.5671578
(Conn_Red-6.40522)*(Disp_Blue-6.8328)	0.0150926	0.000149	101.02	<.0001*	1.2404564
(Conn_Red-6.40522)*(Pow_Blue-9.38597)	0.0599983	0.000353	169.91	<.0001*	1.1784109
(Conn_Red-6.40522)*(Stab_Red-5.84111)	-0.039912	0.000278	-143.3	<.0001*	3.1197283
(Conn_Blue-6.40249)*(Disp_Blue-6.8328)	0.0098679	0.000205	48.19	<.0001*	3.1952372
(Conn_Blue-6.40249)*(Stab_Blue-5.8402)	0.0363405	0.000277	131.31	<.0001*	3.0875662
(Disp_Red-6.83277)*(Stab_Red-5.84111)	0.0111855	0.000319	35.06	<.0001*	2.5365931
(Disp_Blue-6.8328)*(Pow_Blue-9.38597)	-0.077001	0.000504	-152.8	<.0001*	1.7266199
(Disp_Blue-6.8328)*(Stab_Blue-5.8402)	-0.008731	0.00032	-27.28	<.0001*	2.5514926
(Disp_Red-6.83277)*(Disp_Red-6.83277)	0.0020777	0.000158	13.14	<.0001*	1.509493
(Disp_Blue-6.8328)*(Disp_Blue-6.8328)	0.0010165	0.000162	6.27	<.0001*	1.5846471
(Pow_Blue-9.38597)*(Pow_Blue-9.38597)	0.0927675	0.000733	126.60	<.0001*	1.4281583
(Stab_Red-5.84111)*(Stab_Red-5.84111)	-0.000705	0.000358	-1.97	0.0489*	2.5393584
(Stab_Blue-5.8402)*(Stab_Blue-5.8402)	-0.00171	0.000357	-4.79	<.0001*	2.5265606

Figure 36. Final Non Linear Regression Results for Evenly-Matched Engagements

More insights about the reason behind the currently-defined metrics not explaining a substantial amount of variation in the case of unbalanced configurations doing symmetric battle engagements can be found in the next section.

4.8 The Impact of Configuration Sizes on the Variability of Metrics

This research simulated unbalanced forces (X-Y-Z type configurations) doing symmetric engagements (battling each other). For instance, a RED 7-3-4 force would battle a BLUE 7-3-4 force. The only difference between those unbalanced forces is the manner in which the Sensors and Influencers are distributed across Deciders. Figure 37 gives some descriptive statistics of RED percentage win. The most relevant statistic to the impact of symmetric engagements is the mean, which is around 50%. On average, any particular RED configuration has a 50% chance of defeating another BLUE configuration of the same size but a different arrangement of assets. This begs the question of the nature of the impact of configurations of the same size on the performance metrics.

Fundamentally, having same size configurations means there is little variability in the performance metrics among forces. This also means the variability of the percentage win is reduced. Look at the histogram in Figure 38 and the Cumulative Distribution Function chart in Figure 39. The normal distribution and the S-shaped curve reflect the fact that most of the battles were evenly-matched to some extent. This is not counter-intuitive. The regression results for high percentage win engagements, both linear and non-linear showed some performance metrics are significant and some interaction effects are significant as well. This means that a force can indeed be organized for better networked effects. However, the analysis for the evenly-matched engagements, which

represent the majority of engagements, show that in the case of unbalanced configurations doing symmetric engagements, having the best arrangement of assets in terms of the currently-defined metrics will not always provide a decisive advantage. This observed relationship between configuration sizes and the variability of performance metrics explain the fact that 61% and 58% of variation in RED percentage win was not explained by the current metrics for evenly-matched engagements. This also echoes the theme that more research is needed to uncover other potentially significant metrics out there.

Quantiles

100.0%	maximum	65.1
99.5%		57
97.5%		55
90.0%		53.1
75.0%	quartile	51.5
50.0%	median	50
25.0%	quartile	48.4
10.0%		47
2.5%		45.1
0.5%		43.1
0.0%	minimum	36.1

Moments

Mean	50.001174
Std Dev	2.4640661
Std Err Mean	0.0012768
Upper 95% Mean	50.003676
Lower 95% Mean	49.998671
N	3724225

Figure 37. Descriptive Statistics for RED Percentage Win

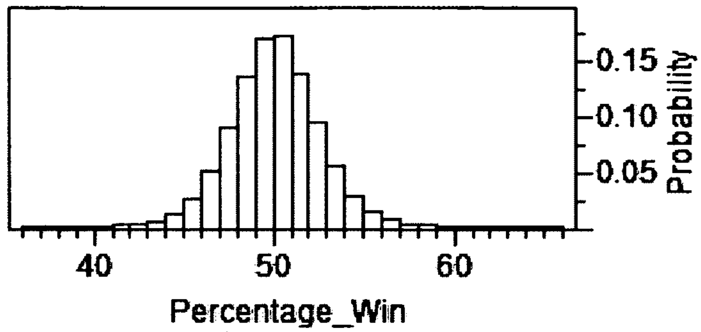

Figure 38. Histogram of RED Percentage Win

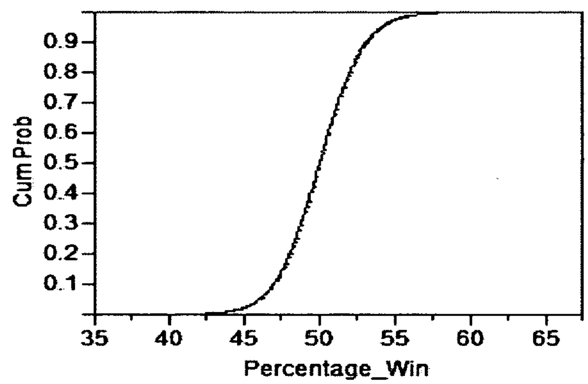

Figure 39. Cumulative Distribution Function Chart of RED Percentage Win

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions and Final Remarks

The objective of this research was to simulate unbalanced force configurations doing symmetric engagements, and study the impact of having purposeful arrangement of assets on the force's combat effectiveness. Another objective was to test the efficacy of discrete-event simulation as a potentially a more efficient and faster modeling tool, with the purpose being allowing more configurations to be run at reasonable time and computer processing costs.

The first objective was carried out successfully. Unbalanced force configurations were simulated doing symmetric engagements, and the data was analyzed to study the impact of networked effects as represented in the performance metrics used in this research. Linear and nonlinear regression analyses were carried out on the aggregated data. Moreover, linear and non-linear regression analyses were conducted after the data was disaggregated to study high percentage win engagements and evenly-matched engagements, respectively. The VIF, which is a measure of multicollinearity, was used to highlight and eliminate metrics that exhibited strong collinearity relationships. The remaining significant metrics were highlighted and given. It was found that a force can be indeed organized for better networked effects as shown in the analysis for high percentage win engagements. However, the analysis for evenly-matched engagements, which represent the majority of engagements simulated in this research, showed that those metrics do not tell the whole story and that having the best arrangement of assets in

terms of the currently-defined metrics will not always provide a decisive advantage. Therefore, more research needs to be done to highlight additional metrics that might wield significant predictive power to help explain the percentage of variation that was not explained by the metrics utilized in this research and previous research attempts. The need for further research into additional metrics is also important because the analyses and an examination of the relationship between configuration sizes and the variability of the metrics shows that the variability in the metrics is somewhat reduced due to the nature of having configurations of the same size battling each other.

The second objective, which is putting the discrete event simulation paradigm to work to build a faster and more efficient model, was also carried out successfully. This research was able to run 285 configurations. This is a substantial increase over previous research attempts that modeled the information age combat model. The time and computer processing costs for the increased experimentation were not at all significant. This research streamlined the whole data collection and analysis process by a complete package of computer programs that calculate the performance metrics, and have them outputted in a format that is friendly to any main-stream statistical software package that future researchers might be interested in using.

5.2 Future Research

This research is the first attempt of its kind to study unbalanced force configurations doing symmetric engagements. It established the groundwork for future research by building a faster and more efficient tool while testing previously-defined

performance metrics. This, however, is merely the beginning, and there is a lot of potential for future research. Among the suggested future research directions are:

- The inclusion of links that connect assets connected to different deciders (i.e. horizontal links) as shown in Table 1. The incorporation of some of those links are more relevant to agent-based modeling because they require some of the complexity which discrete-event simulation took out. However, other types of links, such as a Decider commanding another Decider are a viable extension to this research. What if a fellow Decider, who is a part of an effective combat cycle, intervened in a non-functional combat cycle of another Decider, who had its sensors eliminated? This Decider can use the Sensing assets available to another Decider to form a functional combat cycle. This would add real-world complexity to the model.
- Building a probabilistic model. This would also add a real-world complexity to the model. The assumptions of the simulation developed in this research are that when a target is acquired, it will most certainly be taken out. Another assumption is that at each round of battle, when a random number is generated to decide who takes the next shot, a target will always be sensed. Incorporating probabilities for the Sensing and Influencing functions of a combat force would constitute a very valuable addition.
- Making Deciders a target. This research and previous research attempts have made the simplifying assumption that a Decider cannot be targeted. A Decider, due to its functionality and tasks, would be considered a high-value target in military operations. Making Deciders targetable could potentially have a

sweeping impact on the dynamics of the simulation and the conclusions inferred from the results.

- Incorporating a measurement of how close the battle was. The current output exclusively reports the probability of winning a particular engagement without details regarding it. For instance, it would be very useful to collect data regarding how many Sensors and Influencers are still functional in the winning configuration. It would also be interesting to determine how many functional combat cycles are still present in the winning configuration. A measurement of the "brutality" of the battle constitutes very valuable knowledge that is necessary to augment recommendations on how to properly organize a force to maximize networked effects.
- Running simulations for unbalanced configurations with asymmetric engagements. This research talked about the costly computer technology requirements for pursuing research of that nature in Section 3.4. Extending the discrete event combat model to run unbalanced configurations doing asymmetric engagement will give valuable knowledge relating to the importance of a Sensor and an Influencer to a combat force. It will answer questions such as which type of asset is more important to a combat force. It will also be interesting to see how well do Sensor-intensive configurations perform against Influencer-intensive configurations.
- Calculating eigenvalues: this research decided to discard eigenvalues as a metric due to its diminishing value as a distinguishing factor as the number of unique combinations for a particular configuration becomes larger and larger. However,

it would still be useful to calculate the eigenvalues in future research studies for the sake of completeness, in order to formally assess its predictive power as a metric, even if this power was shown to be insignificant due to the aforementioned reasons.

REFERENCES

Cebrowski, A. K. and Garstka, J. H. (1998). Network Centric Warfare – Its Origin and Future. In *U.S. Naval Institute Proceedings*, pp. 28-35.

Alberts, D. S., and Hayes, R.E. (2003). Power to the Edge. CCRP, Washington, DC.

Coakley, T. P. (1991). Command and Control for War and Peace. National Defense University Press, Washington, DC.

Keohane, R. O., & Nye, J. S. (1998). Power and Interdependence in the Information Age. *Foreign Affairs*. *77*(5), 81-94.

Arquilla, J. (1994). The Strategic Implications of Information Dominance. *Strategic Review*, *22*(3), 24-30.

Nye, J. S. (2004a). Soft power: The means to success in world politics. New York: Public Affairs.

Nye, J. S. (2004b) The Decline of America's Soft Power: Why Washington Should Worry, *Foreign Affairs*, *83*(3), 16-20.

Nye, J. S., and Owens W. A. (1996), America's Information Edge, *Foreign Affairs*, *75*(2), 20-36.

Shakespeare, W. (1954). As you like it. New Haven: Yale University Press.

Mulgan, G. (1998). Connexity: How to live in a connected world. Boston, Mass: Harvard Business School Press.

Harknett, R. J. (1996). Information warfare and deterrence. Parameters. *US Army War College Quarterly*, Autumn, 93-107.

Deller, S. (2009). Towards the quantitative analysis of the connectivity value of networked operations. Ph.D. dissertation, Old Dominion University, United States --

Virginia. Retrieved February 20, 2011, from Dissertations & Theses @ Old Dominion University. (Publication No. AAT 3377889).

Perry, W., Signori, D., and Boon, J. (2004). Exploring Information Superiority: A Methodology for Measuring the Quality of Information and its Impact on Shared Awareness. Santa Monica, CA: The RAND Corporation. ISBN 0-8330-3489-8. Available online at http://www.rand.org/publications/MR/MR1467/.

Oder, J. E. (1994). Digitizing the Battlefield: The Army's First Step to Force XXI, Army.

Arquilla, J., and Ronfeldt, D. (1993). Cyberwar is Coming. *Comparative Strategy, 12*(2), 141-165.

Arquilla, J., and Ronfeldt, D. (1996). The Advent of Netwar, RAND National Defense Research Institute, Santa Monica, CA.

Arquilla, J., and Ronfeldt, D. (1998). The Zapatista Netwar in Mexico, RAND National Defense Research Institute, Santa Monica, CA.

Arquilla, J.; D. Ronfeldt; and M. Zanini. (1999). Networks, Netwar, and Information-Age Terrorism. In Ian O. Lesser et al. (eds.), Countering the New Terrorism, Santa Monica, CA: RAND

Arquilla, J., and Ronfeldt, D. (2000). Swarming & the Future of Conflict, RAND National Defense Research Institute, Santa Monica, CA.

Arquilla, J, and Ronfeldt, D. (2001). Networks and Netwars, RAND National Defense Research Institute, Santa Monica, CA.

Whine, M. (1999). Islamist organizations on the Internet, *Terrorism and Political Violence, 11*(1), 123-132.

Arquilla, J., Ronfeldt, D., & Zanini, M. (1999). Networks, netwar and information age terrorism. In A. Lesser, B. Hoffman, J. Arquilla, D. Ronfeldt, & M. Zanini (Eds.), Countering the new terrorism (pp. 39–84), Santa Monica, CA: RAND.

Pfaltzgraff, R. L., and Shultz, R. H. (1997). War in the information age: New challenges for U.S. security policy, Washington D.C.: Brassey's.

Pardew, J. W. Jr. (1991). The Iraqi Army's Defeat in Kuwait, *Parameters, 21*(4), 17-23;

Toffler, A., and Toffler, H. (1993). War and anti-war: Survival at the dawn of the 21st century, Boston: Little, Brown.

Starry, D. (1983). To Change an Army, *Military Review*, *63*(3), pp. 20-27.

Starry, D. (1997). Extending the battlefield, *Military Review*, *77*(1), 151. Retrieved from EBSCOhost.

Toffler, A. (1980). The Third Wave, New York: Morrow.

Porter, C. D. (2004). Network centric warfare: Transforming the U.S. Army. Carlisle Barracks, Pa: U.S. Army War College.

Miller, S. (2005). Evaluation of Information Assurance Requirements in a Net-Centric Army. Carlisle Barracks, Pa: U.S. Army War College.

Metz, T. F., Garrett, M. W., Hutton, J. E., and Bush, T. W. (2006). Massing Effects in the Information Domain. *Military Review*, 103-113. Retrieved from EBSCOhost.

Murdock, P. (2002). Principles of War on the Network-Centric Battlefield: Mass and Economy of Force. *Parameters: U.S. Army War College, 32*(1), p. 86. Retrieved from EBSCOhost.

Hubenko Jr., V. P., Raines, R. A., Mills, R. F., Baldwin, R. O., Mullins, B. E., & Grimaila, M. R. (2006). Improving the Global Information Grid's Performance through

Satellite Communications Layer Enhancements. *IEEE Communications Magazine, 44*(11), pp. 66-72. Retrieved from EBSCOhost.

Tranchemontagne, M. G. (2001). Terrorist Application of Operational Art. Unpublished Research Paper, U.S. Naval War College, Newport, RI.

Moffat, J. (2003). Complexity theory and network centric warfare. Washington, DC: CCRP Publication Series.

Gonzales, D., Johnson, M., McEver, J., Leedom, D., Kingston, G., and Tseng, M. (2005). Network-centric operations case study, the stryker brigade combat team. Santa Monica, CA: RAND Corporation.

Fidanci, M. (2010). The influence of network factors on network centric operations. Ph.D. dissertation, Old Dominion University, United States -- Virginia. Retrieved February 20, 2011, from Dissertations & Theses @ Old Dominion University. (Publication No. AAT 3429551).

Ling, M., Moon, T., and Kruzins, E. (2005). Proposed Network Centric Warfare Metrics: From Connectivity to the OODA Cycle, Military Operations Research, 10: 5-13.

Fewell, M. P., and Hazen, M. G. (2003) Network-centric warfare—its nature and modelling, Research report DSTO-RR-0262 of the Defence Science and Technology Organisation.

Cares, J. (2005). Distributed Networked Operations. iUniverse, New York.

Jain, S., and Krishna, S. (1998). Autocatalytic Sets and the Growth of Complexity in an Evolutionary Model. *Physical Review Letters*, 81: pp. 5684-5687.

Wilenski, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo.

Earnest, D. C. (2008). Coordination in Large Numbers: An Agent-Based Model of International Negotiations. *International Studies Quarterly*, 52(2), pp. 363-382.

Damaceanu, R. (2008). An agent-based computational study of wealth distribution in function of resource growth interval using NetLogo. *Applied Mathematics & Computation*, 201(1/2), pp. 371-377.

Khasawneh, M. T., Zhang, J., Shearer, N. E., Jaradat, R. M., Rodriguez-Velasquez, E., Bowling, S. R., (2009). An Exploratory Study of the Butterfly Effect Using Agent-Based Modeling, Proceedings of MODSIM World Conference and Expo, Virginia Beach, Virginia.

Siebers, P.-O., Aickelin, U., Celia, H., & Clegg, C. (2010). Simulating Customer Experience and Word-Of-Mouth in Retail - A Case Study. *Simulation*, 86(1), pp. 5-30.

Shendarkar, A., Vasudevan, K., Lee, S., and Son, Y. (2008). Crowd simulation for emergency response using BDI agents based on immersive virtual reality. *Simulation* Modelling Practice & Theory, 16(9), pp. 1415-1429.

Wang, S., Liu, S., & Wang, W. (2008). The simulated impact of RFID-enabled supply chain on pull-based inventory replenishment in TFT-LCD industry. *International Journal of Production Economics*, 112(2), 570-586.

Robinson, S. (2004). Simulation: The Practice of Model Development and Use. Wiley: Chichester, UK.

Banks, J. (2005). Discrete-event system simulation. Upper Saddle River, NJ: Pearson Prentice Hall.

Fishman, G. S. (2001). Discrete-event simulation: modelling, programming, and analysis. Berlin, Springer-Verlag.

Pooch, U. W., and Wall, J. A. (1993). Discrete event simulation: A practical approach. Boca Raton: CRC Press.

Shearer, N.E.N. (2012). A Discrete Event Simulation of Balanced Forces Configurations in Asymmetric Engagements. Doctoral Dissertation in Progress. Old Dominion University, Norfolk, VA.

APPENDICES

APPENDIX A: UNIQUE COMBINATIONS EXPLAINED

To understand what a unique combination is in the context of this research, take a look at the two combinations in Figure 40 below.

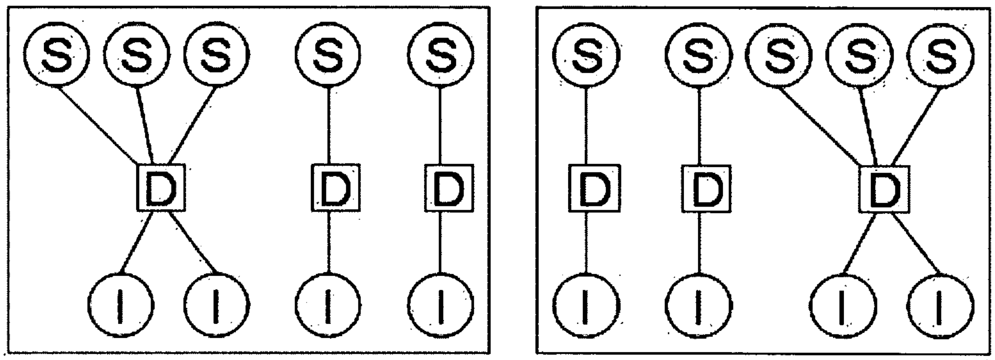

Figure 40. Two Redundant 5-3-4 Combinations

The distribution of Sensors and Influencers across Deciders is tabulated in Table 13 for more clarity.

Table 13. Distribution of Sensors and Influencers across Deciders

No.	D1 Sensors	D2 Sensors	D3 Sensors	D1 Influencers	D2 Influencers	D3 Influencers
1	3	1	1	2	1	1
2	1	1	3	1	1	2

Looking at Figure 40, those two combinations might look different, but they are actually identical. D1, D2, and D3 are just labels the model gives to the Deciders, and whether it is D1 or D3 that has 3 Sensors and 2 Influencers does not really mean anything in the

combat model. As we mentioned earlier, the order is not important. The combat effectiveness and performance on the battlefield will be identical for the two combinations in Figure 40. Only if the distribution of Sensors and Influencers change there would be a difference in the combat performance of two combinations of the same configuration, as in Figure 41 for example. In Figure 41, it is not the order that was changed; it is the distribution of Sensors and Influencers

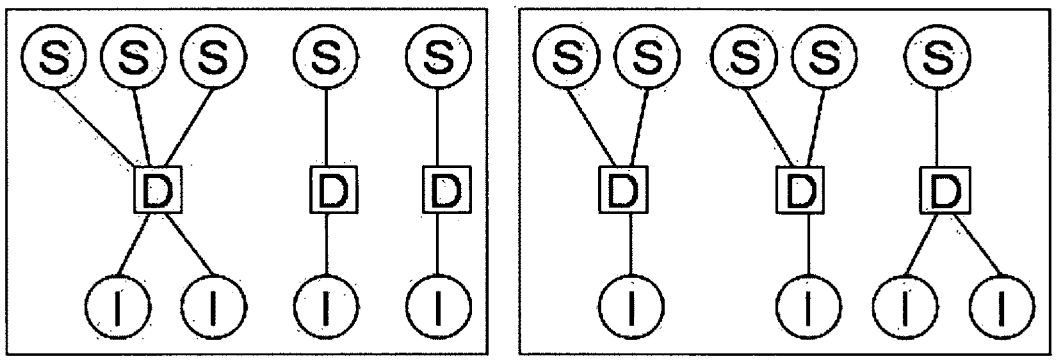

Figure 41. Two Unique 5-3-4 Combinations

Table 14 in Appendix D shows the displays the number of unique combinations for the 285 X-Y-Z configurations that were ran in this research.

APPENDIX B: FIND UNIQUE COMBINATIONS FOR COMBAT CONFIGURATIONS

```
Private Sub Combinations_Click()

Open "D:\OutputXYZ.dat" For Output As #4

upperbound = 11
For i = 3 To upperbound
    For j = i To upperbound
        For k = i To upperbound

'Count number of rows in file to determine X-Y array size
    x = 0
    FileName1 = "D:\Partitions\Partitions_" & j & "_" & i &
".dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        Input #1, Test
        x = x + 1
        Loop
    RowsXY = x / i
    Close #1

 'Count number of rows in file to determine Y-Z array size
    x = 0
    FileName2 = "D:\Partitions\Partitions_" & k & "_" & i &
".dat"

        Open FileName2 For Input As #2    ' Open file for input.
        Do While Not EOF(2)
        Input #2, Test
        x = x + 1
        Loop

    RowsYZ = x / i
    Close #2

Text1 = "XYZ " & j & i & k & " XY " & j & i & " " & RowsXY & " YZ
" & k & i & " " & RowsYZ
Text1.Refresh

'Dimension XY Array to proper size and load data from file
ReDim PartitionXY(RowsXY, i)

        x = 0
        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
```

```
        Input #1, Temp
        n = Int(x / i) + 1
        m = x Mod i + 1
        PartitionXY(n, m) = Temp

        x = x + 1
        Loop
        Close #1

'Dimension YZ Array to proper size and load data from file
ReDim PartitionYZ(RowsYZ, i)

        x = 0
        Open FileName2 For Input As #2    ' Open file for input.
        Do While Not EOF(2)

        Input #2, Temp
        n = Int(x / i) + 1
        m = x Mod i + 1
        PartitionYZ(n, m) = Temp

        x = x + 1
        Loop
        Close #2

'Load new array with evaluation function to determine uniqueness
TotalRows = RowsXY * RowsYZ
ReDim Unique(TotalRows)

'Create unique column from matrix

        s = 1
        Unique(s) = 0
        For h = 1 To RowsXY
            For g = 1 To RowsYZ
                Unique(s) = 0
                For f = 1 To i

        Unique(s) = Round(Unique(s) + PartitionXY(h, f) ^ 3 /
PartitionYZ(g, f) ^ (1 / 3), 6)

                Next f
                s = s + 1

            Next g
        Next h

'Start of new program
ReDim UniqueTemp(1)
Row1 = 2 * i
ReDim UniqueData(Row1, 1)
```

```
Flag = 0
x = 1
UniqueTemp(1) = Unique(1)

'Write first config to array in first column
For a = 1 To i

    UniqueData(2 * (a - 1) + 1, 1) = PartitionXY(1, a)
    UniqueData(2 * a, 1) = PartitionYZ(1, a)

Next a

For q = 2 To TotalRows
For w = 1 To x

If Unique(q) = UniqueTemp(w) Then
Flag = 1
End If

Next w

    If Flag = 0 Then

        x = x + 1
        ReDim Preserve UniqueTemp(x)
        ReDim Preserve UniqueData(Row1, x)
        UniqueTemp(x) = Unique(q)

        place1 = Int((q / RowsYZ) - 0.00001) + 1
        place2 = q Mod RowsYZ
        If place2 = 0 Then place2 = RowsYZ

'For-Next Loop to iterate writing of row data

        For b = 1 To i
            UniqueData(2 * (b - 1) + 1, x) = PartitionXY(place1,
b)
            UniqueData(2 * b, x) = PartitionYZ(place2, b)

        Next b

    End If

Flag = 0

Next q

'Print i, j, x
Print #4, j & " " & i & " " & k & " " & x

'Write Files
```

```
FileName3 = "D:\PartitionDataM\PartitionM_" & j & "_" & i & "_" &
k & ".dat"

Open FileName3 For Output As #3

'Write Raw Data to files
Config = ""
For u = 1 To x

    For r = 1 To i
        Config = Config & UniqueData(2 * (r - 1) + 1, u) & " " &
UniqueData(2 * r, u) & " "
    Next r

Print #3, Config

Config = ""
Next u

Close #3

        Next k
    Next j
Next i

End Sub
```

APPENDIX C: COUNT AND LIST ALL COMBAT CONFIGURATIONS

```
Private Sub Combinations_Click()

upperbound = 11

Open "D:\XYZConfig.dat" For Output As #1
x = 0

For i = 3 To upperbound
    For j = i To upperbound
        For k = i To upperbound

        x = x + 1
        Print #1, x & " " & j & " " & i & " " & k

        Next k
    Next j
Next i

Close #1

Print "Done"

End Sub
```

APPENDIX D: FIND TOTAL NUMBER OF UNIQUE SYMMETRIC ENGAGEMENTS

```
Private Sub Combinations_Click()

Open "D:\DataMahmoud.dat" For Output As #3
For upperbound = 3 To 11
configcount = 0

Open "D:\XYZConfig.dat" For Output As #1
x = 0

For i = 3 To upperbound
    For j = i To upperbound
        For k = i To upperbound

    y = 0
    FileName1 = "D:\PartitionDataM\PartitionM_" & j & "_" & i &
"_" & k & ".dat"

        Open FileName1 For Input As #2    ' Open file for input.
        Do While Not EOF(2)
        Input #2, Test
        y = y + 1
        Loop

    Rows = y / (2 * i)
    Close #2

        x = x + 1

        Print #1, x & " " & j & " " & i & " " & k & " " & Rows

configcount = configcount + Rows ^ 2

        Next k
    Next j
Next i

Close #1

Print #3, x & " " & configcount

Next upperbound
Print "Done"

End Sub
```

Table 14. Number of Unique Combinations for All X-Y-Z Configurations

Count	Configuration	Unique Engagements
1	3-3-3	1
2	3-3-4	1
3	3-3-5	4
4	3-3-6	9
5	3-3-7	16
6	3-3-8	25
7	3-3-9	49
8	3-3-10	64
9	3-3-11	100
10	4-3-3	1
11	4-3-4	4
12	4-3-5	16
13	4-3-6	36
14	4-3-7	81
15	4-3-8	144
16	4-3-9	256
17	4-3-10	400
18	4-3-11	625
19	5-3-3	4
20	5-3-4	16
21	5-3-5	64
22	5-3-6	144
23	5-3-7	324
24	5-3-8	576
25	5-3-9	1,024
26	5-3-10	1,600
27	5-3-11	2,500
28	6-3-3	9
29	6-3-4	36
30	6-3-5	144
31	6-3-6	361
32	6-3-7	784
33	6-3-8	1,444
34	6-3-9	2,601
35	6-3-10	4,096
36	6-3-11	6,400
37	7-3-3	16
38	7-3-4	81
39	7-3-5	324
40	7-3-6	784
41	7-3-7	1,764
42	7-3-8	3,249
43	7-3-9	5,776
44	7-3-10	9,216
45	7-3-11	14,400

46	8-3-3	25
47	8-3-4	144
48	8-3-5	576
49	8-3-6	1,444
50	8-3-7	3,249
51	8-3-8	6,084
52	8-3-9	10,816
53	8-3-10	17,424
54	8-3-11	27,225
55	9-3-3	49
56	9-3-4	256
57	9-3-5	1,024
58	9-3-6	2,601
59	9-3-7	5,776
60	9-3-8	10,816
61	9-3-9	19,321
62	9-3-10	30,976
63	9-3-11	48,400
64	10-3-3	64
65	10-3-4	400
66	10-3-5	1,600
67	10-3-6	4,096
68	10-3-7	9,216
69	10-3-8	17,424
70	10-3-9	30,976
71	10-3-10	50,176
72	10-3-11	78,400
73	11-3-3	81
74	11-3-4	625
75	11-3-5	2,500
76	11-3-6	6,400
77	11-3-7	14,400
78	11-3-8	27,225
79	11-3-9	47,961
80	11-3-10	78,400
81	11-3-11	122,500
82	4-4-4	1
83	4-4-5	1
84	4-4-6	4
85	4-4-7	9
86	4-4-8	25
87	4-4-9	36
88	4-4-10	81
89	4-4-11	121
90	5-4-4	1
91	5-4-5	4
92	5-4-6	16
93	5-4-7	49

94	5-4-8	121
95	5-4-9	256
96	5-4-10	529
97	5-4-11	961
98	6-4-4	4
99	6-4-5	16
100	6-4-6	81
101	6-4-7	225
102	6-4-8	625
103	6-4-9	1,296
104	6-4-10	2,809
105	6-4-11	5,041
106	7-4-4	9
107	7-4-5	49
108	7-4-6	225
109	7-4-7	729
110	7-4-8	1,936
111	7-4-9	4,356
112	7-4-10	9,216
113	7-4-11	17,424
114	8-4-4	25
115	8-4-5	121
116	8-4-6	625
117	8-4-7	1,936
118	8-4-8	5,476
119	8-4-9	12,100
120	8-4-10	26,244
121	8-4-11	49,284
122	9-4-4	36
123	9-4-5	256
124	9-4-6	1,296
125	9-4-7	4,356
126	9-4-8	12,100
127	9-4-9	28,224
128	9-4-10	60,516
129	9-4-11	116,964
130	10-4-4	81
131	10-4-5	529
132	10-4-6	2,809
133	10-4-7	9,216
134	10-4-8	26,244
135	10-4-9	60,516
136	10-4-10	131,769
137	10-4-11	253,009
138	11-4-4	121
139	11-4-5	961
140	11-4-6	5,041
141	11-4-7	17,424

142	11-4-8	49,284
143	11-4-9	116,964
144	11-4-10	253,009
145	11-4-11	494,209
146	5-5-5	1
147	5-5-6	1
148	5-5-7	4
149	5-5-8	9
150	5-5-9	25
151	5-5-10	49
152	5-5-11	100
153	6-5-5	1
154	6-5-6	4
155	6-5-7	16
156	6-5-8	49
157	6-5-9	144
158	6-5-10	324
159	6-5-11	729
160	7-5-5	4
161	7-5-6	16
162	7-5-7	81
163	7-5-8	256
164	7-5-9	784
165	7-5-10	1,849
166	7-5-11	4,356
167	8-5-5	9
168	8-5-6	49
169	8-5-7	256
170	8-5-8	900
171	8-5-9	2,809
172	8-5-10	7,056
173	8-5-11	16,900
174	9-5-5	25
175	9-5-6	144
176	9-5-7	784
177	9-5-8	2,809
178	9-5-9	9,025
179	9-5-10	23,104
180	9-5-11	56,169
181	10-5-5	49
182	10-5-6	324
183	10-5-7	1,849
184	10-5-8	7,056
185	10-5-9	23,104
186	10-5-10	61,504
187	10-5-11	151,321
188	11-5-5	100
189	11-5-6	729

190	11-5-7	4,356
191	11-5-8	16,900
192	11-5-9	56,169
193	11-5-10	151,321
194	11-5-11	376,996
195	6-6-6	1
196	6-6-7	1
197	6-6-8	4
198	6-6-9	9
199	6-6-10	25
200	6-6-11	49
201	7-6-6	1
202	7-6-7	4
203	7-6-8	16
204	7-6-9	49
205	7-6-10	144
206	7-6-11	361
207	8-6-6	4
208	8-6-7	16
209	8-6-8	81
210	8-6-9	256
211	8-6-10	841
212	8-6-11	2,116
213	9-6-6	9
214	9-6-7	49
215	9-6-8	256
216	9-6-9	961
217	9-6-10	3,136
218	9-6-11	8,649
219	10-6-6	25
220	10-6-7	144
221	10-6-8	841
222	10-6-9	3,136
223	10-6-10	11,025
224	10-6-11	30,625
225	11-6-6	49
226	11-6-7	361
227	11-6-8	2,116
228	11-6-9	8,649
229	11-6-10	30,625
230	11-6-11	90,000
231	7-7-7	1
232	7-7-8	1
233	7-7-9	4
234	7-7-10	9
235	7-7-11	25
236	8-7-7	1
237	8-7-8	4

238	8-7-9	16
239	8-7-10	49
240	8-7-11	144
241	9-7-7	4
242	9-7-8	16
243	9-7-9	81
244	9-7-10	256
245	9-7-11	841
246	10-7-7	9
247	10-7-8	49
248	10-7-9	256
249	10-7-10	961
250	10-7-11	3,249
251	11-7-7	25
252	11-7-8	144
253	11-7-9	841
254	11-7-10	3,249
255	11-7-11	11,664
256	8-8-8	1
257	8-8-9	1
258	8-8-10	4
259	8-8-11	9
260	9-8-8	1
261	9-8-9	4
262	9-8-10	16
263	9-8-11	49
264	10-8-8	4
265	10-8-9	16
266	10-8-10	81
267	10-8-11	256
268	11-8-8	9
269	11-8-9	49
270	11-8-10	256
271	11-8-11	961
272	9-9-9	1
273	9-9-10	1
274	9-9-11	4
275	10-9-9	1
276	10-9-10	4
277	10-9-11	16
278	11-9-9	4
279	11-9-10	16
280	11-9-11	81
281	10-10-10	1
282	10-10-11	1
283	11-10-10	1
284	11-10-11	4
285	11-11-11	1

APPENDIX E: FIND NUMBER OF UNIQUE ASYMMETRIC ENGAGEMENTS

```
Private Sub Combinations_Click()

Open "D:\Data.dat" For Output As #3
For upperbound = 3 To 15
configcount = 0

Open "D:\XYZConfig.dat" For Output As #1
X = 0

For I = 3 To upperbound
    For j = I To upperbound
        For k = I To upperbound

    Y = 0
    FileName1 = "D:\PartitionDataM\PartitionM_" & j & "_" & I &
"_" & k & ".dat"

        Open FileName1 For Input As #2   ' Open file for input.
        Do While Not EOF(2)
        Input #2, Test
        Y = Y + 1
        Loop

    Rows = Y / (2 * I)
    Close #2

        X = X + 1

        Print #1, X & " " & j & " " & I & " " & k & " " & Rows

        Next k
    Next j
Next I

Close #1

ReDim CountArray(X)
Z = 0

Open "D:\XYZConfig.dat" For Input As #1
Do While Not EOF(1)
Z = Z + 1
Input #1, Countholder, S, D, I, RowsCount
CountArray(Z) = RowsCount
Loop

Close #1
```

```
Counttotalrows = 0
For g = 1 To X
    For h = g To X
Counttotalrows1 = CountArray(g) * CountArray(h)
Counttotalrows = Counttotalrows + Counttotalrows1
    Next h
Next g

Print #3, X & " " & Counttotalrows

Next upperbound
Print "Done"
Close #3

End Sub
```

APPENDIX F: PROPOSED DISCRETE EVENT COMBAT MODEL

```
Dim TempNodes, RedX, BlueX, TempSensors, TempDeciders
Dim RedPX, RedPY, BluePX, BluePY

Public Sub Combat_Click()
Randomize
Cls

Open "D:\XYZConfig.dat" For Input As #4

Open "D:\output.dat" For Output As #2

Counttotal = 0

'Determine number of replications below

Replications = 1000

'Determine Lower and Upper bounds below according to your
segregation rationale

lowerbound = 1
upperbound = 285

40        Do While Not EOF(4)
            Input #4, CountConfig, RedPX, RedPY, RedPZ
            If CountConfig >= lowerbound And CountConfig <=
upperbound Then

BluePX = RedPX
BluePY = RedPY
BluePZ = RedPZ

UpdateRX = CountConfig & " " & "Red " & RedPX & "-" & RedPY & "-"
& RedPZ & " vs Blue " & BluePX & "-" & BluePY & "-" & BluePZ
UpdateRX.Refresh
Form1.Show

'Load Red Source Matrix to Array from file

a = RedPX
b = RedPY
c = RedPZ

'Count number of rows in file to determine array size

    x = 0
    FileName1 = "D:\PartitionDataM\PartitionM_" & a & "_" & b &
"_" & c & ".dat"
```

```
        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        Input #1, Test
        x = x + 1
        Loop

    RowsRed = x / (2 * b)
    Close #1

TempNodes = 0
RedY = b
TotalRed = RowsRed
i = 1

'Print RowsRed

ReDim Red(TotalRed, RedY, 2)
ReDim RedTemp(TotalRed, RedY, 2)

Open FileName1 For Input As #1    ' Open file for input.
Do While Not EOF(1)
TempNodes = TempNodes + 1
Input #1, TempSensors, TempDeciders
Red(i, TempNodes, 1) = TempSensors
RedTemp(i, TempNodes, 1) = TempSensors
Red(i, TempNodes, 2) = TempDeciders
RedTemp(i, TempNodes, 2) = TempDeciders
    If TempNodes = RedY Then
        TempNodes = 0
        i = i + 1
    End If

Loop
Close #1

'Load Blue Source Matrix to Array from file

BluePX = RedPX
BluePY = RedPY
BluePZ = RedPZ

d = BluePX
e = BluePY
f = BluePZ

'Count number of rows in file to determine array size

    x = 0
    FileName2 = "D:\PartitionDataM\PartitionM_" & d & "_" & e &
"_" & f & ".dat"
```

```
        Open FileName2 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        Input #1, Test
        x = x + 1
        Loop

    RowsBlue = x / (2 * e)
    Close #1

TempNodes = 0
BlueY = e
TotalBlue = RowsBlue
i = 1

ReDim Blue(TotalBlue, BlueY, 2)
ReDim BlueTemp(TotalBlue, BlueY, 2)

Open FileName2 For Input As #1    ' Open file for input.
Do While Not EOF(1)
TempNodes = TempNodes + 1
Input #1, TempSensors, TempDeciders
Blue(i, TempNodes, 1) = TempSensors
BlueTemp(i, TempNodes, 1) = TempSensors
Blue(i, TempNodes, 2) = TempDeciders
BlueTemp(i, TempNodes, 2) = TempDeciders
    If TempNodes = BlueY Then
        TempNodes = 0
        i = i + 1
    End If

Loop
Close #1

'Do Battle

CountRep = 0
RedCount = 0
Do While RedCount < TotalRed
RedCount = RedCount + 1
BlueCount = 0
Do While BlueCount < TotalBlue
BlueCount = BlueCount + 1
CountRep = CountRep + 1
Counttotal = Counttotal + 1

'Determine Number of Replications (e.g. 30)

RedWins = 0
Bluewins = 0
Do While RedWins + Bluewins < Replications

'Need to reinitialize the matrix each time
```

```
'Load Red Source Matrix from Initial Temp Matrix

For i = 1 To TotalRed
For j = 1 To RedY
Red(i, j, 1) = RedTemp(i, j, 1)
Red(i, j, 2) = RedTemp(i, j, 2)
Next j
Next i

'Load Blue Source Matrix from Initial Temp Matrix

For i = 1 To TotalBlue
For j = 1 To BlueY
Blue(i, j, 1) = BlueTemp(i, j, 1)
Blue(i, j, 2) = BlueTemp(i, j, 2)
Next j
Next i

'Determine winner of each replication

Winner = ""
Do While Winner = ""

'Count Red Sensors and Influencers and Combat Cycles

TotalActiveRedSensors = 0
TotalActiveRedInfluencers = 0
TotalActiveRedCombatCycles = 0
For i = 1 To RedY
    RedFlagS = 0
    RedFlagI = 0
    TotalActiveRedSensors = TotalActiveRedSensors + Red(RedCount,
i, 1)
    TotalActiveRedInfluencers    =    TotalActiveRedInfluencers    +
Red(RedCount, i, 2)
    If Red(RedCount, i, 1) > 0 Then RedFlagS = 1
    If Red(RedCount, i, 2) > 0 Then RedFlagI = 1
    TotalActiveRedCombatCycles    =    TotalActiveRedCombatCycles    +
RedFlagS * RedFlagI
Next i

If TotalActiveRedCombatCycles = 0 Then
Bluewins = Bluewins + 1
'Print "Blue Wins"
Winner = "Blue"

GoTo 10
End If

'Count Blue Sensors and Influencers and Combat Cycles

TotalActiveBlueSensors = 0
```

```
TotalActiveBlueInfluencers = 0
TotalActiveBlueCombatCycles = 0
For i = 1 To BlueY
    BlueFlagS = 0
    BlueFlagI = 0
    TotalActiveBlueSensors     =     TotalActiveBlueSensors     +
Blue(BlueCount, i, 1)
    TotalActiveBlueInfluencers   =   TotalActiveBlueInfluencers   +
Blue(BlueCount, i, 2)
    If Blue(BlueCount, i, 1) > 0 Then BlueFlagS = 1
    If Blue(BlueCount, i, 2) > 0 Then BlueFlagI = 1
    TotalActiveBlueCombatCycles  =  TotalActiveBlueCombatCycles  +
BlueFlagS * BlueFlagI
Next i

If TotalActiveBlueCombatCycles = 0 Then
RedWins = RedWins + 1
'Print "Red Wins"
Winner = "Red"

GoTo 10
End If

'Pick Side to Shoot and Destory Sensor or Influencer on Opposing
Side

TotalActiveEverything       =       TotalActiveRedSensors       +
TotalActiveBlueSensors
ShootSide = Rnd() * TotalActiveEverything

If ShootSide <= TotalActiveRedSensors Then

'Red won toss so destroy Blue target (sensor or influencer)

    BlueDestroy   =   Int(Rnd   *   (TotalActiveBlueSensors   +
TotalActiveBlueInfluencers)) + 1
    BlueTrack = 0
    For j = 1 To 2
    For i = 1 To BlueY
        BlueTrack = BlueTrack + Blue(BlueCount, i, j)
        If BlueTrack >= BlueDestroy Then
        Blue(BlueCount, i, j) = Blue(BlueCount, i, j) - 1
        GoTo 20
        End If
    Next i
    Next j

Else

'Blue won toss so destroy Red target (sensor or influencer)
```

```
    RedDestroy    =    Int(Rnd    *    (TotalActiveRedSensors    +
TotalActiveRedInfluencers)) + 1
    RedTrack = 0
    For j = 1 To 2
    For i = 1 To RedY
        RedTrack = RedTrack + Red(RedCount, i, j)
        If RedTrack >= RedDestroy Then
        Red(RedCount, i, j) = Red(RedCount, i, j) - 1
        GoTo 20
        End If
    Next i
    Next j

End If

20
Loop
10
Loop

HolderRed = ""
HolderBlue = ""
For p = 1 To b

HolderRed = HolderRed & " " & RedTemp(RedCount, p, 1) & " " &
RedTemp(RedCount, p, 2)

Next p

For p = 1 To b

HolderBlue = HolderBlue & " " & BlueTemp(BlueCount, p, 1) & " " &
BlueTemp(BlueCount, p, 2)

Next p

Print #2, CountConfig & " " & Counttotal & " " & CountRep & " " &
RedPX & " " & RedPY & " " & RedPZ & " " & BluePX & " " & BluePY &
" " & BluePZ & " " & HolderRed & " , " & HolderBlue & " " &
Round((100 * RedWins / (RedWins + Bluewins)), 2)

Loop
Loop

            Else: GoTo 40
        End If
Loop


Close #2
Close #4
```

```
Print CountRep, "Done"
End Sub
```

APPENDIX G: PROGRAM TO CALCULATE DISPARITY

```
Private Sub Combinations_Click()
CounterIndex = 0
Open "D:\Disparity1000.dat" For Output As #2

    FileName1 = "D:\Mahmoud1000Reps.dat"

' Mahmoud1000Reps.dat is the raw data file.

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i


        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics

        MaxSensor = RedConfig(1)
        MinSensor = RedConfig(1)
        MaxInfluencer = RedConfig(2)
        MinInfluencer = RedConfig(2)

        For i = 1 To (2 * RedI - 1)


        If RedConfig(i) > MaxSensor Then MaxSensor = RedConfig(i)
        If RedConfig(i) < MinSensor Then MinSensor = RedConfig(i)

        If RedConfig(i + 1) > MaxInfluencer Then MaxInfluencer =
RedConfig(i + 1)
        If RedConfig(i + 1) < MinInfluencer Then MinInfluencer =
RedConfig(i + 1)
```

```
        RedDisparity = (MaxSensor - MinSensor) + (MaxInfluencer -
MinInfluencer)

        i = i + 1
        Next i

        MaxSensor = BlueConfig(1)
        MinSensor = BlueConfig(1)
        MaxInfluencer = BlueConfig(2)
        MinInfluencer = BlueConfig(2)

        For i = 1 To (2 * BlueI - 1)

        If BlueConfig(i) > MaxSensor Then MaxSensor =
BlueConfig(i)
        If BlueConfig(i) < MinSensor Then MinSensor =
BlueConfig(i)

        If BlueConfig(i + 1) > MaxInfluencer Then MaxInfluencer =
BlueConfig(i + 1)
        If BlueConfig(i + 1) < MinInfluencer Then MinInfluencer =
BlueConfig(i + 1)

        BlueDisparity = (MaxSensor - MinSensor) + (MaxInfluencer
- MinInfluencer)
        i = i + 1
        Next i

        Print #2, CounterIndex, RedDisparity, BlueDisparity,
PercentWin

        Loop

    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX H: PROGRAM TO CALCULATE ROBUSTNESS

```
Private Sub Combinations_Click()
CounterIndex = 0
Open "D:\Robustness1000.dat" For Output As #2

    FileName1 = "D:\Mahmoud1000Reps.dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i

        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics
        RedRobustness = 0
        BlueRobustness = 0

        For i = 1 To (2 * RedI - 1)

        RedRobustnessTemp = RedConfig(i)

        If RedConfig(i + 1) < RedConfig(i) Then RedRobustnessTemp
= RedConfig(i + 1)

        RedRobustness = RedRobustness + RedRobustnessTemp

        i = i + 1
        Next i

        For i = 1 To (2 * BlueI - 1)

        BlueRobustnessTemp = BlueConfig(i)
```

```
        If BlueConfig(i + 1) < BlueConfig(i) Then
BlueRobustnessTemp = BlueConfig(i + 1)

        BlueRobustness = BlueRobustness + BlueRobustnessTemp

        i = i + 1
        Next i

        Print #2, CounterIndex, RedRobustness, BlueRobustness,
Round(PercentWin, 2)

        Loop

    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX I: PROGRAM TO CALCULATE STRENGTH

```
Private Sub Combinations_Click()

Open "D:\Strength1000.dat" For Output As #2
CounterIndex = 0

    FileName1 = "D:\Mahmoud1000Reps.dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i

        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics

        RedStrength = 0
        For i = 1 To (2 * RedI - 1)
            RedStrength = RedStrength + (Log(RedConfig(i) + 1) /
Log(10)) * (Log(RedConfig(i + 1) + 1) / Log(10))
        i = i + 1
        Next i

        BlueStrength = 0
        For i = 1 To (2 * BlueI - 1)
            BlueStrength = BlueStrength + (Log(BlueConfig(i) + 1)
/ Log(10)) * (Log(BlueConfig(i + 1) + 1) / Log(10))
        i = i + 1
        Next i

        Print #2, CounterIndex, Round(RedStrength, 2),
Round(BlueStrength, 2), Round(PercentWin, 2)
```

```
        Loop

    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX J: PROGRAM TO CALCULATE POWER

```
Private Sub Combinations_Click()
CounterIndex = 0
Open "D:\Power1000.dat" For Output As #2

    FileName1 = "D:\Mahmoud1000Reps.dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i

        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics

        RedPower = 0
        For i = 1 To (2 * RedI - 1)
            RedPower = RedPower + RedConfig(i) ^ 0.5 *
RedConfig(i + 1) ^ 0.5
        i = i + 1
        Next i

        BluePower = 0
        For i = 1 To (2 * BlueI - 1)
            BluePower = BluePower + BlueConfig(i) ^ 0.5 *
BlueConfig(i + 1) ^ 0.5
        i = i + 1
        Next i

        Print #2, CounterIndex, Round(RedPower, 2),
Round(BluePower, 2), Round(PercentWin, 2)


        Loop
```

```
    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX K: PROGRAM TO CALCULATE STABILITY

```
Private Sub Combinations_Click()
CounterIndex = 0
Open "D:\Stability1000.dat" For Output As #2

    FileName1 = "D:\Mahmoud1000Reps.dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i

        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics

        RedStability = 0
        For i = 1 To (2 * RedI - 1)
            RedStability = RedStability + RedConfig(i) /
RedConfig(i + 1)
        i = i + 1
        Next i

        BlueStability = 0
        For i = 1 To (2 * BlueI - 1)
            BlueStability = BlueStability + BlueConfig(i) /
BlueConfig(i + 1)
        i = i + 1
        Next i

        Print #2, CounterIndex, Round(RedStability, 2),
Round(BlueStability, 2), Round(PercentWin, 2)


        Loop
```

```
    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX L: PROGRAM TO CALCULATE CONNECTIVITY

```
Private Sub Combinations_Click()
CounterIndex = 0
Open "D:\Connectivity1000.dat" For Output As #2

    FileName1 = "D:\Mahmoud1000Reps.dat"

        Open FileName1 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        CounterIndex = CounterIndex + 1
        Input #1, IndexCount
        Input #1, TotalCount
        Input #1, ConfigCount
        Input #1, RedS, RedI, RedD, BlueS, BlueI, BlueD

        ReDim RedConfig(2 * RedI)
        ReDim BlueConfig(2 * BlueI)

        For i = 1 To 2 * RedI
        Input #1, RedConfig(i)
        Next i

        For i = 1 To 2 * BlueI
        Input #1, BlueConfig(i)
        Next i

        Input #1, PercentWin

'Calculate Metrics

        RedConnectivity = 0
        For i = 1 To (2 * RedI - 1)
            RedConnectivity = RedConnectivity + Abs(RedConfig(i)
- RedConfig(i + 1))
        i = i + 1
        Next i

        BlueConnectivity = 0
        For i = 1 To (2 * BlueI - 1)
            BlueConnectivity = BlueConnectivity +
Abs(BlueConfig(i) - BlueConfig(i + 1))
        i = i + 1
        Next i

        Print #2, CounterIndex, Round(RedConnectivity, 2),
Round(BlueConnectivity, 2), Round(PercentWin, 2)


        Loop
```

```
    Print "Done"

      Close #1
      Close #2

End Sub
```

APPENDIX M: ADJUSTED COMBAT MODEL FOR TIMEKEEPING

```
Dim TempNodes, RedX, BlueX, TempSensors, TempDeciders
Dim RedPX, RedPY, BluePX, BluePY

Public Sub Combat_Click()
Randomize
Cls
Dim StartTime As Double, EndTime As Double
StartTime = Timer

Open "D:\XYZConfig.dat" For Input As #4

Open "D:\TIME-OUTPUT.dat" For Output As #2

Counttotal = 0
Replications = 30
lowerbound = 194
upperbound = 194

40         Do While Not EOF(4)
             Input #4, CountConfig, RedPX, RedPY, RedPZ
             If CountConfig >= lowerbound And CountConfig <=
upperbound Then

BluePX = RedPX
BluePY = RedPY
BluePZ = RedPZ

UpdateRX = CountConfig & " " & "Red " & RedPX & "-" & RedPY & "-"
& RedPZ & " vs Blue " & BluePX & "-" & BluePY & "-" & BluePZ
UpdateRX.Refresh
Form1.Show

'Load Red Source Matrix to Array from file

a = RedPX
b = RedPY
c = RedPZ

'Count number of rows in file to determine array size
    x = 0
    FileName1 = "D:\PartitionDataM\PartitionM_" & a & "_" & b &
"_" & c & ".dat"

        Open FileName1 For Input As #1   ' Open file for input.
        Do While Not EOF(1)
        Input #1, Test
        x = x + 1
        Loop
```

```
    RowsRed = x / (2 * b)
    Close #1

TempNodes = 0
RedY = b
TotalRed = RowsRed
i = 1

'Print RowsRed

ReDim Red(TotalRed, RedY, 2)
ReDim RedTemp(TotalRed, RedY, 2)

Open FileName1 For Input As #1    ' Open file for input.
Do While Not EOF(1)
TempNodes = TempNodes + 1
Input #1, TempSensors, TempDeciders
Red(i, TempNodes, 1) = TempSensors
RedTemp(i, TempNodes, 1) = TempSensors
Red(i, TempNodes, 2) = TempDeciders
RedTemp(i, TempNodes, 2) = TempDeciders
    If TempNodes = RedY Then
        TempNodes = 0
        i = i + 1
    End If

Loop
Close #1

'Load Blue Source Matrix to Array from file

BluePX = RedPX
BluePY = RedPY
BluePZ = RedPZ

d = BluePX
e = BluePY
f = BluePZ

'Count number of rows in file to determine array size
    x = 0
    FileName2 = "D:\PartitionDataM\PartitionM_" & d & "_" & e &
"_" & f & ".dat"

        Open FileName2 For Input As #1    ' Open file for input.
        Do While Not EOF(1)
        Input #1, Test
        x = x + 1
        Loop

    RowsBlue = x / (2 * e)
```

```
    Close #1

TempNodes = 0
BlueY = e
TotalBlue = RowsBlue
i = 1

ReDim Blue(TotalBlue, BlueY, 2)
ReDim BlueTemp(TotalBlue, BlueY, 2)

Open FileName2 For Input As #1   ' Open file for input.
Do While Not EOF(1)
TempNodes = TempNodes + 1
Input #1, TempSensors, TempDeciders
Blue(i, TempNodes, 1) = TempSensors
BlueTemp(i, TempNodes, 1) = TempSensors
Blue(i, TempNodes, 2) = TempDeciders
BlueTemp(i, TempNodes, 2) = TempDeciders
    If TempNodes = BlueY Then
        TempNodes = 0
        i = i + 1
    End If

Loop
Close #1

'Do Battle

CountRep = 0
RedCount = 0
Do While RedCount < TotalRed
RedCount = RedCount + 1
BlueCount = 0
Do While BlueCount < TotalBlue
BlueCount = BlueCount + 1
CountRep = CountRep + 1
Counttotal = Counttotal + 1

'Determine Number of Replications (e.g. 30)
RedWins = 0
Bluewins = 0
Do While RedWins + Bluewins < Replications

'Need to reinitialize the matrix each time

'Load Red Source Matrix from Initial Temp Matrix

For i = 1 To TotalRed
For j = 1 To RedY
Red(i, j, 1) = RedTemp(i, j, 1)
Red(i, j, 2) = RedTemp(i, j, 2)
Next j
```

```
Next i

'Load Blue Source Matrix from Initial Temp Matrix
For i = 1 To TotalBlue
For j = 1 To BlueY
Blue(i, j, 1) = BlueTemp(i, j, 1)
Blue(i, j, 2) = BlueTemp(i, j, 2)
Next j
Next i

'Determine winner of each replication
Winner = ""
Do While Winner = ""

'Count Red Sensors and Influencers and Combat Cycles
TotalActiveRedSensors = 0
TotalActiveRedInfluencers = 0
TotalActiveRedCombatCycles = 0
For i = 1 To RedY
    RedFlagS = 0
    RedFlagI = 0
    TotalActiveRedSensors = TotalActiveRedSensors + Red(RedCount,
i, 1)
    TotalActiveRedInfluencers = TotalActiveRedInfluencers +
Red(RedCount, i, 2)
    If Red(RedCount, i, 1) > 0 Then RedFlagS = 1
    If Red(RedCount, i, 2) > 0 Then RedFlagI = 1
    TotalActiveRedCombatCycles = TotalActiveRedCombatCycles +
RedFlagS * RedFlagI
Next i

If TotalActiveRedCombatCycles = 0 Then
Bluewins = Bluewins + 1
'Print "Blue Wins"
Winner = "Blue"

GoTo 10
End If

'Count Blue Sensors and Influencers and Combat Cycles
TotalActiveBlueSensors = 0
TotalActiveBlueInfluencers = 0
TotalActiveBlueCombatCycles = 0
For i = 1 To BlueY
    BlueFlagS = 0
    BlueFlagI = 0
    TotalActiveBlueSensors = TotalActiveBlueSensors +
Blue(BlueCount, i, 1)
    TotalActiveBlueInfluencers = TotalActiveBlueInfluencers +
Blue(BlueCount, i, 2)
    If Blue(BlueCount, i, 1) > 0 Then BlueFlagS = 1
    If Blue(BlueCount, i, 2) > 0 Then BlueFlagI = 1
```

```
    TotalActiveBlueCombatCycles = TotalActiveBlueCombatCycles +
BlueFlagS * BlueFlagI
Next i

If TotalActiveBlueCombatCycles = 0 Then
RedWins = RedWins + 1
'Print "Red Wins"
Winner = "Red"

GoTo 10
End If

'Pick Side to Shoot and Destory Sensor or Influencer on Opposing
Side

TotalActiveEverything = TotalActiveRedSensors +
TotalActiveBlueSensors

'ShootSide = Int(Rnd() * TotalActiveEverything) + 1
ShootSide = Rnd() * TotalActiveEverything

If ShootSide <= TotalActiveRedSensors Then
'Red won toss so destroy Blue target (sensor or influencer)
    BlueDestroy = Int(Rnd * (TotalActiveBlueSensors +
TotalActiveBlueInfluencers)) + 1
    BlueTrack = 0
    For j = 1 To 2
    For i = 1 To BlueY
        BlueTrack = BlueTrack + Blue(BlueCount, i, j)
        If BlueTrack >= BlueDestroy Then
        Blue(BlueCount, i, j) = Blue(BlueCount, i, j) - 1
        GoTo 20
        End If
    Next i
    Next j

Else

'Blue won toss so destroy Red target (sensor or influencer)
    RedDestroy = Int(Rnd * (TotalActiveRedSensors +
TotalActiveRedInfluencers)) + 1
    RedTrack = 0
    For j = 1 To 2
    For i = 1 To RedY
        RedTrack = RedTrack + Red(RedCount, i, j)
        If RedTrack >= RedDestroy Then
        Red(RedCount, i, j) = Red(RedCount, i, j) - 1
        GoTo 20
        End If
    Next i
    Next j
```

```
End If

20
Loop
10
Loop

HolderRed = ""
HolderBlue = ""
For p = 1 To b

HolderRed = HolderRed & " " & RedTemp(RedCount, p, 1) & " " &
RedTemp(RedCount, p, 2)

Next p

For p = 1 To b

HolderBlue = HolderBlue & " " & BlueTemp(BlueCount, p, 1) & " " &
BlueTemp(BlueCount, p, 2)

Next p
EndTime = Timer
Print #2, CountConfig & " " & Counttotal & " " & CountRep & " " &
RedPX & " " & RedPY & " " & RedPZ & " " & BluePX & " " & BluePY &
" " & BluePZ & " " & HolderRed & " , " & HolderBlue & " " &
Round((100 * RedWins / (RedWins + Bluewins)), 2) & " " & EndTime
- StartTime

Loop
Loop

            Else: GoTo 40
        End If
Loop


Close #2
Close #4

Print CountRep, "Done"
End Sub
```

APPENDIX N: COMPUTER SPECIFICATIONS

Table 15 below shows the profile the Belarc Advisor program created for one of the computers used in running the simulations for this research.

Table 15. Profile for Computers Used to Run Simulations

Operating System	System Model
Windows XP Professional Service Pack 3 (build 2600) Install Language: English (United States) System Locale: English (United States) Installed: 1/11/2007 1:36:15 PM	Gateway E6610 System Serial Number: 0040292933 Chassis Serial Number: 0040292933 Enclosure Type: Desktop
Processor	**Main Circuit Board**
2.40 gigahertz Intel Core 2 Duo 64 kilobyte primary memory cache 4096 kilobyte secondary memory cache 64-bit ready Multi-core (2 total) Not hyper-threaded	Board: Intel Corporation OEMD975XLAG1 AAD50908-205 Serial Number: BQLA727001TG Bus Clock: 266 megahertz BIOS: Intel Corp. LA97510J.15A.0285.2007.0906.0226 09/06/2007

APPENDIX O: SUPLLEMENTAL TABLES TO DATA ANALYSIS

Table 16. Parameter Estimates for Regression Accounting for Quadratic Terms and 2-Way Interactions

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept	Biased	50.006862	0.01567	3191.3	<.0001*
Conn_Red	Biased	-0.085982	0.004756	-18.08	<.0001*
Conn_Blue	Biased	0.0723843	0.001316	55.01	<.0001*
Disp_Red	Biased	0.038256	0.000838	45.65	<.0001*
Disp_Blue	Biased	-0.035594	0.000838	-42.48	<.0001*
Pow_Red	Biased	2.8861726	0.018312	157.61	<.0001*
Pow_Blue	Biased	-2.877947	0.018312	-157.2	<.0001*
Rob_Red	Biased	-0.028626	0.011016	-2.60	0.0094*
Rob_Blue	Zeroed	0	0	.	.
Stab_Red	Biased	0.1310243	0.001095	119.65	<.0001*
Stab_Blue	Biased	-0.125477	0.001095	-114.6	<.0001*
Stre_Red	Biased	-6.271722	0.114447	-54.80	<.0001*
Stre_Blue	Biased	6.4152626	0.114447	56.05	<.0001*
(Conn_Red-6.49002)*(Conn_Blue-6.49002)	Biased	-0.009159	0.00574	-1.60	0.1106
(Conn_Red-6.49002)*(Disp_Red-6.912)	Biased	-0.021815	0.002763	-7.90	<.0001*
(Conn_Red-6.49002)*(Disp_Blue-6.912)	Biased	0.0294096	0.002839	10.36	<.0001*
(Conn_Red-6.49002)*(Pow_Red-9.3553)	Biased	-0.458053	100.8178	-0.00	0.9964
(Conn_Red-6.49002)*(Pow_Blue-9.3553)	Biased	-1.368375	126.1184	-0.01	0.9913
(Conn_Red-6.49002)*(Rob_Red-6.95869)	Biased	0.2037027	47.02409	0.00	0.9965
(Conn_Red-6.49002)*(Rob_Blue-6.95869)	Biased	0.0083422	0.017669	0.47	0.6368
(Conn_Red-6.49002)*(Stab_Red-5.87047)	Biased	-0.038642	0.003653	-10.58	<.0001*
(Conn_Red-6.49002)*(Stab_Blue-5.87047)	Biased	0.0398489	0.003859	10.33	<.0001*
(Conn_Red-6.49002)*(Stre_Red-1.04305)	Biased	1.2871527	0.282406	4.56	<.0001*
(Conn_Red-6.49002)*(Stre_Blue-1.04305)	Biased	15.15843	1400.849	0.01	0.9914
(Conn_Blue-6.49002)*(Disp_Red-6.912)	Biased	-0.005112	0.000769	-6.64	<.0001*
(Conn_Blue-6.49002)*(Disp_Blue-6.912)	Biased	0.001757	0.000796	2.21	0.0273*
(Conn_Blue-6.49002)*(Pow_Red-9.3553)	Biased	0.3980703	100.8178	0.00	0.9968
(Conn_Blue-6.49002)*(Pow_Blue-9.3553)	Biased	1.4699667	126.1184	0.01	0.9907
(Conn_Blue-6.49002)*(Rob_Red-6.95869)	Biased	-0.244245	47.02409	-0.01	0.9959
(Conn_Blue-6.49002)*(Rob_Blue-6.95869)	Biased	-0.024041	0.001638	-14.67	<.0001*
(Conn_Blue-6.49002)*(Stab_Red-5.87047)	Biased	0.0021993	0.001044	2.11	0.0351*
(Conn_Blue-6.49002)*(Stab_Blue-5.87047)	Biased	-0.003496	0.001177	-2.97	0.0030*
(Conn_Blue-6.49002)*(Stre_Red-1.04305)	Biased	-0.618656	0.080158	-7.72	<.0001*
(Conn_Blue-6.49002)*(Stre_Blue-1.04305)	Biased	-15.79348	1400.849	-0.01	0.9910
(Disp_Red-6.912)*(Disp_Blue-6.912)	Biased	-0.000122	0.000658	-0.18	0.8533
(Disp_Red-6.912)*(Pow_Red-9.3553)	Biased	-0.065699	0.010381	-6.33	<.0001*
(Disp_Red-6.912)*(Pow_Blue-9.3553)	Biased	0.1228488	0.010675	11.51	<.0001*
(Disp_Red-6.912)*(Rob_Red-6.95869)	Biased	-0.036025	0.006429	-5.60	<.0001*
(Disp_Red-6.912)*(Rob_Blue-6.95869)	Zeroed	0	0	.	.
(Disp_Red-6.912)*(Stab_Red-5.87047)	Biased	0.0134665	0.000808	16.67	<.0001*
(Disp_Red-6.912)*(Stab_Blue-5.87047)	Biased	-0.000396	0.000748	-0.53	0.5962
(Disp_Red-6.912)*(Stre_Red-1.04305)	Biased	0.554937	0.07318	7.58	<.0001*
(Disp_Red-6.912)*(Stre_Blue-1.04305)	Biased	-0.821973	0.072223	-11.38	<.0001*
(Disp_Blue-6.912)*(Pow_Red-9.3553)	Biased	-0.120722	0.010675	-11.31	<.0001*
(Disp_Blue-6.912)*(Pow_Blue-9.3553)	Biased	0.0558277	0.010381	5.38	<.0001*
(Disp_Blue-6.912)*(Rob_Red-6.95869)	Biased	0.0490516	0.006429	7.63	<.0001*
(Disp_Blue-6.912)*(Rob_Blue-6.95869)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Stab_Red-5.87047)	Biased	0.000288	0.000748	0.39	0.7001
(Disp_Blue-6.912)*(Stab_Blue-5.87047)	Biased	-0.013682	0.000808	-16.93	<.0001*
(Disp_Blue-6.912)*(Stre_Red-1.04305)	Biased	0.8144933	0.072223	11.28	<.0001*
(Disp_Blue-6.912)*(Stre_Blue-1.04305)	Biased	-0.569209	0.07318	-7.78	<.0001*
(Pow_Red-9.3553)*(Pow_Blue-9.3553)	Biased	-0.122308	0.168723	-0.72	0.4685
(Pow_Red-9.3553)*(Rob_Red-6.95869)	Biased	-1.010111	201.6356	-0.01	0.9960
(Pow_Red-9.3553)*(Rob_Blue-6.95869)	Biased	0.6152063	201.6356	0.00	0.9976

Term		Estimate	Std Error	t Ratio	Prob>|t|
(Pow_Red-9.3553)*(Stab_Red-5.87047)	Biased	-0.033303	0.011595	-2.87	0.0041*
(Pow_Red-9.3553)*(Stab_Blue-5.87047)	Biased	-0.017213	0.011934	-1.44	0.1492
(Pow_Red-9.3553)*(Stre_Red-1.04305)	Biased	-11.03939	1.24965	-8.83	<.0001*
(Pow_Red-9.3553)*(Stre_Blue-1.04305)	Biased	-0.427597	1.143155	-0.37	0.7084
(Pow_Blue-9.3553)*(Rob_Red-6.95869)	Biased	-2.494151	252.2369	-0.01	0.9921
(Pow_Blue-9.3553)*(Rob_Blue-6.95869)	Biased	3.0310069	252.2369	0.01	0.9904
(Pow_Blue-9.3553)*(Stab_Red-5.87047)	Biased	0.0108896	0.011934	0.91	0.3615
(Pow_Blue-9.3553)*(Stab_Blue-5.87047)	Biased	0.0423826	0.011595	3.66	0.0003*
(Pow_Blue-9.3553)*(Stre_Red-1.04305)	Biased	0.9584633	1.143155	0.84	0.4018
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	Biased	10.251389	1.249652	8.20	<.0001*
(Rob_Red-6.95869)*(Rob_Blue-6.95869)	Biased	-0.440693	94.04819	-0.00	0.9963
(Rob_Red-6.95869)*(Stab_Red-5.87047)	Biased	-0.081765	0.008751	-9.34	<.0001*
(Rob_Red-6.95869)*(Stab_Blue-5.87047)	Biased	0.0821597	0.008751	9.39	<.0001*
(Rob_Red-6.95869)*(Stre_Red-1.04305)	Biased	4.536612	0.656918	6.91	<.0001*
(Rob_Red-6.95869)*(Stre_Blue-1.04305)	Biased	28.839821	2801.697	0.01	0.9918
(Rob_Blue-6.95869)*(Stab_Red-5.87047)	Zeroed	0	0	.	.
(Rob_Blue-6.95869)*(Stab_Blue-5.87047)	Zeroed	0	0	.	.
(Rob_Blue-6.95869)*(Stre_Red-1.04305)	Zeroed	0	0	.	.
(Rob_Blue-6.95869)*(Stre_Blue-1.04305)	Biased	-33.35682	2801.697	-0.01	0.9905
(Stab_Red-5.87047)*(Stab_Blue-5.87047)		-0.001519	0.001119	-1.36	0.1748
(Stab_Red-5.87047)*(Stre_Red-1.04305)	Biased	0.7563111	0.082995	9.11	<.0001*
(Stab_Red-5.87047)*(Stre_Blue-1.04305)	Biased	-0.155434	0.07912	-1.96	0.0495*
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	Biased	0.211902	0.07912	2.68	0.0074*
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	Biased	-0.809198	0.082995	-9.75	<.0001*
(Stre_Red-1.04305)*(Stre_Blue-1.04305)	Biased	0.8741248	7.812463	0.11	0.9109
(Conn_Red-6.49002)*(Conn_Red-6.49002)	Zeroed	0	0	.	.
(Conn_Blue-6.49002)*(Conn_Blue-6.49002)	Zeroed	0	0	.	.
(Disp_Red-6.912)*(Disp_Red-6.912)	Biased	-0.000312	0.000425	-0.73	0.4630
(Disp_Blue-6.912)*(Disp_Blue-6.912)	Biased	-3.665e-5	0.000425	-0.09	0.9312
(Pow_Red-9.3553)*(Pow_Red-9.3553)	Biased	0.672105	0.101012	6.65	<.0001*
(Pow_Blue-9.3553)*(Pow_Blue-9.3553)	Biased	-0.602216	0.101013	-5.96	<.0001*
(Rob_Red-6.95869)*(Rob_Red-6.95869)	Biased	0.3556914	94.04818	0.00	0.9970
(Rob_Blue-6.95869)*(Rob_Blue-6.95869)	Zeroed	0	0	.	.
(Stab_Red-5.87047)*(Stab_Red-5.87047)		0.0060697	0.000789	7.69	<.0001*
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)		-0.003787	0.000789	-4.80	<.0001*
(Stre_Red-1.04305)*(Stre_Red-1.04305)		16.809371	4.336531	3.88	0.0001*
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)		-16.11488	4.33654	-3.72	0.0002*

Table 17. Parameter Estimates for Non Linear Regression Analysis Second Iteration for Aggregated Data

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	49.981865	0.014867	3362.0	<.0001*
Conn_Red	-0.080315	0.004468	-17.97	<.0001*
Conn_Blue	0.0630253	0.001205	52.31	<.0001*
Disp_Red	0.0386345	0.000796	48.56	<.0001*
Disp_Blue	-0.03479	0.000795	-43.75	<.0001*
Pow_Red	2.8139016	0.016849	167.01	<.0001*
Pow_Blue	-2.792098	0.016892	-165.3	<.0001*
Rob_Red	-0.04156	0.010361	-4.01	<.0001*
Stab_Red	0.1352955	0.00102	132.63	<.0001*
Stab_Blue	-0.128844	0.001009	-127.6	<.0001*
Stre_Red	-5.443371	0.104682	-52.00	<.0001*
Stre_Blue	5.5813844	0.104184	53.57	<.0001*
(Conn_Red-6.49002)*(Disp_Red-6.912)	-0.028753	0.00138	-20.84	<.0001*
(Conn_Red-6.49002)*(Disp_Blue-6.912)	0.0322231	0.00129	24.99	<.0001*
(Conn_Red-6.49002)*(Stab_Red-5.87047)	-0.03721	0.001882	-19.78	<.0001*
(Conn_Red-6.49002)*(Stab_Blue-5.87047)	0.0296053	0.001903	15.55	<.0001*
(Conn_Red-6.49002)*(Stre_Red-1.04305)	0.1451616	0.022759	6.38	<.0001*
(Conn_Blue-6.49002)*(Disp_Red-6.912)	0.0016162	0.000451	3.59	0.0003*
(Conn_Blue-6.49002)*(Disp_Blue-6.912)	-0.002112	0.000431	-4.90	<.0001*
(Conn_Blue-6.49002)*(Stab_Red-5.87047)	0.0059021	0.000578	10.21	<.0001*
(Conn_Blue-6.49002)*(Stab_Blue-5.87047)	0.0008796	0.000635	1.38	0.1661
(Conn_Blue-6.49002)*(Stre_Red-1.04305)	0.0462072	0.005574	8.29	<.0001*
(Disp_Red-6.912)*(Pow_Red-9.3553)	-0.06888	0.004204	-16.38	<.0001*
(Disp_Red-6.912)*(Pow_Blue-9.3553)	0.1359794	0.004068	33.43	<.0001*
(Disp_Red-6.912)*(Rob_Red-6.95869)	-0.056054	0.003251	-17.24	<.0001*
(Disp_Red-6.912)*(Stab_Red-5.87047)	0.0158812	0.000439	36.14	<.0001*
(Disp_Red-6.912)*(Stre_Red-1.04305)	0.5967172	0.023016	25.93	<.0001*
(Disp_Red-6.912)*(Stre_Blue-1.04305)	-0.886018	0.022737	-38.97	<.0001*
(Disp_Blue-6.912)*(Pow_Red-9.3553)	-0.135118	0.004088	-33.05	<.0001*
(Disp_Blue-6.912)*(Pow_Blue-9.3553)	0.0564977	0.004091	13.81	<.0001*
(Disp_Blue-6.912)*(Rob_Red-6.95869)	0.0653813	0.003115	20.99	<.0001*
(Disp_Blue-6.912)*(Stab_Blue-5.87047)	-0.014526	0.000439	-33.08	<.0001*
(Disp_Blue-6.912)*(Stre_Red-1.04305)	0.8798619	0.022702	38.76	<.0001*
(Disp_Blue-6.912)*(Stre_Blue-1.04305)	-0.585403	0.022976	-25.48	<.0001*
(Pow_Red-9.3553)*(Stab_Red-5.87047)	-0.083436	0.006268	-13.31	<.0001*
(Pow_Red-9.3553)*(Stre_Red-1.04305)	-2.701181	0.417751	-6.47	<.0001*
(Pow_Blue-9.3553)*(Stab_Blue-5.87047)	0.0723799	0.006109	11.85	<.0001*
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	1.6110981	0.411876	3.91	<.0001*
(Rob_Red-6.95869)*(Stab_Red-5.87047)	-0.058585	0.004591	-12.76	<.0001*
(Rob_Red-6.95869)*(Stab_Blue-5.87047)	0.0600855	0.004282	14.03	<.0001*
(Rob_Red-6.95869)*(Stre_Red-1.04305)	0.6277513	0.057202	10.97	<.0001*
(Stab_Red-5.87047)*(Stre_Red-1.04305)	0.9412939	0.043365	21.71	<.0001*
(Stab_Red-5.87047)*(Stre_Blue-1.04305)	-0.016169	0.017173	-0.94	0.3464
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	0.1086323	0.017494	6.21	<.0001*
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	-0.943993	0.043203	-21.85	<.0001*
(Pow_Red-9.3553)*(Pow_Red-9.3553)	0.0446184	0.027818	1.60	0.1087
(Pow_Blue-9.3553)*(Pow_Blue-9.3553)	-0.022163	0.027823	-0.80	0.4257
(Stab_Red-5.87047)*(Stab_Red-5.87047)	0.0081631	0.000545	14.97	<.0001*
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)	-0.005508	0.000544	-10.12	<.0001*
(Stre_Red-1.04305)*(Stre_Red-1.04305)	-5.111218	1.59414	-3.21	0.0013*
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)	7.796369	1.592166	4.90	<.0001*

Table 18. Parameter Estimates for Non Linear Regression Analysis Third Iteration for Aggregated Data

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	49.983591	0.014768	3384.5	<.0001*	.
Conn_Red	-0.079537	0.004442	-17.91	<.0001*	243.22375
Conn_Blue	0.0626174	0.001181	53.01	<.0001*	17.201585
Disp_Red	0.0386102	0.000795	48.57	<.0001*	4.3388881
Disp_Blue	-0.034894	0.000788	-44.28	<.0001*	4.2629389
Pow_Red	2.8132055	0.016814	167.32	<.0001*	376.44501
Pow_Blue	-2.79236	0.01685	-165.7	<.0001*	378.05549
Rob_Red	-0.039955	0.010313	-3.87	0.0001*	366.78072
Stab_Red	0.1351153	0.001015	133.14	<.0001*	4.8938095
Stab_Blue	-0.128637	0.000992	-129.7	<.0001*	4.6776274
Stre_Red	-5.44366	0.104354	-52.17	<.0001*	261.80138
Stre_Blue	5.5770939	0.10396	53.65	<.0001*	259.83047
(Conn_Red-6.49002)*(Disp_Red-6.912)	-0.028379	0.001356	-20.93	<.0001*	175.37863
(Conn_Red-6.49002)*(Disp_Blue-6.912)	0.0317772	0.001252	25.39	<.0001*	100.35974
(Conn_Red-6.49002)*(Stab_Red-5.87047)	-0.038597	0.001398	-27.62	<.0001*	92.631186
(Conn_Red-6.49002)*(Stab_Blue-5.87047)	0.0316776	0.001228	25.79	<.0001*	61.156856
(Conn_Red-6.49002)*(Stre_Red-1.04305)	0.1411118	0.022635	6.23	<.0001*	126.76952
(Conn_Blue-6.49002)*(Disp_Red-6.912)	0.0014707	0.000436	3.38	0.0007*	12.150939
(Conn_Blue-6.49002)*(Disp_Blue-6.912)	-0.001894	0.00041	-4.62	<.0001*	16.027593
(Conn_Blue-6.49002)*(Stab_Red-5.87047)	0.0064424	0.000283	22.77	<.0001*	3.2444923
(Conn_Blue-6.49002)*(Stre_Red-1.04305)	0.0478034	0.005483	8.72	<.0001*	6.1630102
(Disp_Red-6.912)*(Pow_Red-9.3553)	-0.06641	0.003963	-16.76	<.0001*	133.06619
(Disp_Red-6.912)*(Pow_Blue-9.3553)	0.1349546	0.003897	34.63	<.0001*	104.3635
(Disp_Red-6.912)*(Rob_Red-6.95869)	-0.05516	0.003208	-17.19	<.0001*	252.64791
(Disp_Red-6.912)*(Stab_Red-5.87047)	0.0157578	0.00043	36.63	<.0001*	5.4586934
(Disp_Red-6.912)*(Stre_Red-1.04305)	0.5787881	0.020265	28.56	<.0001*	62.41219
(Disp_Red-6.912)*(Stre_Blue-1.04305)	-0.884266	0.021955	-40.28	<.0001*	61.003902
(Disp_Blue-6.912)*(Pow_Red-9.3553)	-0.136974	0.003986	-34.36	<.0001*	109.1913
(Disp_Blue-6.912)*(Pow_Blue-9.3553)	0.0566987	0.003744	15.14	<.0001*	118.7969
(Disp_Blue-6.912)*(Rob_Red-6.95869)	0.0645115	0.003058	21.10	<.0001*	165.9083
(Disp_Blue-6.912)*(Stab_Blue-5.87047)	-0.014459	0.000424	-34.10	<.0001*	5.3017035
(Disp_Blue-6.912)*(Stre_Red-1.04305)	0.8896733	0.021959	40.51	<.0001*	61.025682
(Disp_Blue-6.912)*(Stre_Blue-1.04305)	-0.576911	0.020226	-28.52	<.0001*	62.167767
(Pow_Red-9.3553)*(Stab_Red-5.87047)	-0.07681	0.005306	-14.48	<.0001*	149.73009
(Pow_Red-9.3553)*(Stre_Red-1.04305)	-2.041491	0.086126	-23.70	<.0001*	354.48798
(Pow_Blue-9.3553)*(Stab_Blue-5.87047)	0.0692297	0.00478	14.48	<.0001*	121.52102
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	1.3162078	0.073794	17.84	<.0001*	260.24522
(Rob_Red-6.95869)*(Stab_Red-5.87047)	-0.061496	0.003862	-15.92	<.0001*	186.88454
(Rob_Red-6.95869)*(Stab_Blue-5.87047)	0.064335	0.003181	20.22	<.0001*	121.72936
(Rob_Red-6.95869)*(Stre_Red-1.04305)	0.6182766	0.056963	10.85	<.0001*	281.00762
(Stab_Red-5.87047)*(Stre_Red-1.04305)	0.8937715	0.035932	24.87	<.0001*	124.74185
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	0.1051904	0.017326	6.07	<.0001*	27.113337
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	-0.946869	0.036738	-25.77	<.0001*	130.40228
(Stab_Red-5.87047)*(Stab_Red-5.87047)	0.0079153	0.000523	15.13	<.0001*	5.8780379
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)	-0.005481	0.000523	-10.49	<.0001*	5.8674517
(Stre_Red-1.04305)*(Stre_Red-1.04305)	-7.526219	0.538911	-13.97	<.0001*	239.56066
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)	8.8008061	0.540509	16.28	<.0001*	240.9839

APPENDIX P: REGRESSION RESULTS

This appendix will be used to harbor all the relevant regression output in one place for ease of access to interested readers.

Linear Regression with All Metrics

Singularity Details

Disp_Red = Disp_Blue
Conn_Red = Conn_Blue - 2*Rob_Red + 2*Rob_Blue

Summary of Fit

RSquare	0.548116
RSquare Adj	0.548115
Root Mean Square Error	1.656404
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	12394051	1239405	451732.0
Error	3.7e+6	10218028	2.743674	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
				1.0698
Lack Of Fit	3.5e+6	9740188	2.75262	Prob > F
Pure Error	185705	477840	2.57311	<.0001*
Total Error	3.7e+6	10218028		Max RSq
				0.9789

Parameter Estimates

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept	Biased	49.979922	0.011075	4512.7	<.0001*
Conn_Red	Biased	0.0404894	0.003127	12.95	<.0001*
Conn_Blue	Biased	-0.102744	0.000947	-108.4	<.0001*
Disp_Red	Biased	0.027845	0.000595	46.83	<.0001*
Disp_Blue	Zeroed	0	0	.	.
Pow_Red	Biased	3.7250245	0.009123	408.31	<.0001*
Pow_Blue	Biased	-3.604289	0.010097	-357.0	<.0001*
Rob_Red	Biased	-0.140805	0.007364	-19.12	<.0001*
Rob_Blue	Zeroed	0	0	.	.
Stab_Red		0.1185791	0.000927	127.91	<.0001*
Stab_Blue		-0.103433	0.000905	-114.3	<.0001*
Stre_Red		-5.384312	0.055111	-97.70	<.0001*
Stre_Blue		5.3787573	0.056146	95.80	<.0001*

Linear Regression with All Metrics – After Removing Collinear Relationships

Summary of Fit

RSquare	0.548116
RSquare Adj	0.548115
Root Mean Square Error	1.656404
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	12394051	1239405	451732.0
Error	3.7e+6	10218028	2.743674	**Prob > F**
C. Total	3.7e+6	22612079		<.0001*

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
				1.0698
Lack Of Fit	3.5e+6	9740188	2.75262	**Prob > F**
Pure Error	185705	477840	2.57311	<.0001*
Total Error	3.7e+6	10218028		**Max RSq**
				0.9789

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	49.979922	0.011075	4512.7	<.0001*
Conn_Red	0.0404894	0.003127	12.95	<.0001*
Conn_Blue	-0.102744	0.000947	-108.4	<.0001*
Disp_Red	0.027845	0.000595	46.83	<.0001*
Pow_Red	3.7250245	0.009123	408.31	<.0001*
Pow_Blue	-3.604289	0.010097	-357.0	<.0001*
Rob_Red	-0.140805	0.007364	-19.12	<.0001*
Stab_Red	0.1185791	0.000927	127.91	<.0001*
Stab_Blue	-0.103433	0.000905	-114.3	<.0001*
Stre_Red	-5.384312	0.055111	-97.70	<.0001*
Stre_Blue	5.3787573	0.056146	95.80	<.0001*

Non-Linear Regression with All Metrics

Summary of Fit

RSquare	0.575822
RSquare Adj	0.575813
Root Mean Square Error	1.604837
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	74	13020525	175953	68317.98
Error	3.7e+6	9591553	2.575501	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
				1.0010
Lack Of Fit	3.5e+6	9113713.3	2.57563	Prob > F
Pure Error	185705	477840.1	2.57311	0.3863
Total Error	3.7e+6	9591553.3		Max RSq
				0.9789

Term		Estimate	Std Error	t Ratio	Prob>|t|
Intercept	Biased	49.951983	0.015503	3222.1	<.0001*
Conn_Red	Biased	-0.136667	0.004557	-29.99	<.0001*
Conn_Blue	Biased	0.0817714	0.001275	64.12	<.0001*
Disp_Red	Biased	0.0221688	0.000738	30.05	<.0001*
Disp_Blue	Zeroed	0	0	.	.
Pow_Red	Biased	3.3428626	0.014228	234.95	<.0001*
Pow_Blue	Biased	-3.223306	0.014527	-221.9	<.0001*
Rob_Red	Biased	-0.122739	0.010688	-11.48	<.0001*
Rob_Blue	Zeroed	0	0	.	.
Stab_Red	Biased	0.1279588	0.00108	118.45	<.0001*
Stab_Blue	Biased	-0.116767	0.001069	-109.2	<.0001*
Stre_Red	Biased	-9.362989	0.083009	-112.8	<.0001*
Stre_Blue	Biased	9.2913684	0.082223	113.00	<.0001*
(Conn_Red-6.49002)*(Conn_Blue-6.49002)	Biased	0.025967	0.005209	4.98	<.0001*
(Conn_Red-6.49002)*(Disp_Red-6.912)	Biased	-0.68475	27.01352	-0.03	0.9798
(Conn_Red-6.49002)*(Disp_Blue-6.912)	Zeroed	0	0	.	.
(Conn_Red-6.49002)*(Pow_Red-9.3553)	Biased	-1.455962	104.513	-0.01	0.9889
(Conn_Red-6.49002)*(Pow_Blue-9.3553)	Biased	-0.258146	66.1372	-0.00	0.9969
(Conn_Red-6.49002)*(Rob_Red-6.95869)	Biased	0.5142077	21.31002	0.02	0.9807
(Conn_Red-6.49002)*(Rob_Blue-6.95869)	Biased	0.089486	19.23589	0.00	0.9963
(Conn_Red-6.49002)*(Stab_Red-5.87047)	Biased	-1.935207	77.40924	-0.02	0.9801
(Conn_Red-6.49002)*(Stab_Blue-5.87047)	Biased	0.0625795	62.8163	0.00	0.9992
(Conn_Red-6.49002)*(Stre_Red-1.04305)	Biased	33.105766	863.3145	0.04	0.9694
(Conn_Red-6.49002)*(Stre_Blue-1.04305)	Biased	-3.306769	499.8272	-0.01	0.9947
(Conn_Blue-6.49002)*(Disp_Red-6.912)	Biased	0.6798569	27.01352	0.03	0.9799
(Conn_Blue-6.49002)*(Disp_Blue-6.912)	Zeroed	0	0	.	.
(Conn_Blue-6.49002)*(Pow_Red-9.3553)	Biased	1.0856473	104.513	0.01	0.9917
(Conn_Blue-6.49002)*(Pow_Blue-9.3553)	Biased	0.5054041	66.13717	0.01	0.9939
(Conn_Blue-6.49002)*(Rob_Red-6.95869)	Biased	-0.515236	21.31004	-0.02	0.9807
(Conn_Blue-6.49002)*(Rob_Blue-6.95869)	Biased	0.0036542	19.23588	0.00	0.9998
(Conn_Blue-6.49002)*(Stab_Red-5.87047)	Biased	1.9032667	77.40924	0.02	0.9804
(Conn_Blue-6.49002)*(Stab_Blue-5.87047)	Biased	-0.058316	62.8163	-0.00	0.9993
(Conn_Blue-6.49002)*(Stre_Red-1.04305)	Biased	-29.81424	863.3146	-0.03	0.9725

Term		Estimate	Std Error	t Ratio	Prob>|t|
(Conn_Blue-6.49002)*(Stre_Blue-1.04305)	Biased	0.475803	499.8272	0.00	0.9992
(Disp_Red-6.912)*(Disp_Blue-6.912)	Biased	-0.003627	0.000295	-12.30	<.0001*
(Disp_Red-6.912)*(Pow_Red-9.3553)	Biased	-0.174603	0.006194	-28.19	<.0001*
(Disp_Red-6.912)*(Pow_Blue-9.3553)	Biased	0.1844717	0.006927	26.63	<.0001*
(Disp_Red-6.912)*(Rob_Red-6.95869)	Biased	-1.368943	54.02704	-0.03	0.9798
(Disp_Red-6.912)*(Rob_Blue-6.95869)	Biased	1.3669531	54.02704	0.03	0.9798
(Disp_Red-6.912)*(Stab_Red-5.87047)	Biased	0.0136937	0.000685	19.98	<.0001*
(Disp_Red-6.912)*(Stab_Blue-5.87047)	Biased	-0.005708	0.000619	-9.22	<.0001*
(Disp_Red-6.912)*(Stre_Red-1.04305)	Biased	1.1796733	0.039351	29.98	<.0001*
(Disp_Red-6.912)*(Stre_Blue-1.04305)	Biased	-1.292706	0.037026	-34.91	<.0001*
(Disp_Blue-6.912)*(Pow_Red-9.3553)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Pow_Blue-9.3553)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Rob_Red-6.95869)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Rob_Blue-6.95869)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Stab_Red-5.87047)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Stab_Blue-5.87047)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Stre_Red-1.04305)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Stre_Blue-1.04305)	Zeroed	0	0	.	.
(Pow_Red-9.3553)*(Pow_Blue-9.3553)	Biased	-1.955382	0.102972	-18.99	<.0001*
(Pow_Red-9.3553)*(Rob_Red-6.95869)	Biased	-2.824288	209.026	-0.01	0.9892
(Pow_Red-9.3553)*(Rob_Blue-6.95869)	Biased	2.0081479	209.0261	0.01	0.9923
(Pow_Red-9.3553)*(Stab_Red-5.87047)	Biased	-0.037239	0.009359	-3.98	<.0001*
(Pow_Red-9.3553)*(Stab_Blue-5.87047)	Biased	0.1247335	0.009661	12.91	<.0001*
(Pow_Red-9.3553)*(Stre_Red-1.04305)	Biased	-28.69715	0.706996	-40.59	<.0001*
(Pow_Red-9.3553)*(Stre_Blue-1.04305)	Biased	15.134962	0.539946	28.03	<.0001*
(Pow_Blue-9.3553)*(Rob_Red-6.95869)	Biased	-0.356643	132.2744	-0.00	0.9978
(Pow_Blue-9.3553)*(Rob_Blue-6.95869)	Biased	0.9351264	132.2743	0.01	0.9944
(Pow_Blue-9.3553)*(Stab_Red-5.87047)	Biased	0.0108387	0.010284	1.05	0.2919
(Pow_Blue-9.3553)*(Stab_Blue-5.87047)	Biased	-0.055323	0.010377	-5.33	<.0001*
(Pow_Blue-9.3553)*(Stre_Red-1.04305)	Biased	13.643498	0.65646	20.78	<.0001*
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	Biased	-3.354334	0.733237	-4.57	<.0001*
(Rob_Red-6.95869)*(Rob_Blue-6.95869)	Biased	-0.965419	65.0041	-0.01	0.9882
(Rob_Red-6.95869)*(Stab_Red-5.87047)	Biased	-3.875334	154.8185	-0.03	0.9800
(Rob_Red-6.95869)*(Stab_Blue-5.87047)	Biased	0.1323885	125.6326	0.00	0.9992
(Rob_Red-6.95869)*(Stre_Red-1.04305)	Biased	67.128672	1726.629	0.04	0.9690
(Rob_Red-6.95869)*(Stre_Blue-1.04305)	Biased	-7.329074	999.6545	-0.01	0.9942
(Rob_Blue-6.95869)*(Stab_Red-5.87047)	Biased	3.8024727	154.8185	0.02	0.9804
(Rob_Blue-6.95869)*(Stab_Blue-5.87047)	Biased	-0.120175	125.6326	-0.00	0.9992
(Rob_Blue-6.95869)*(Stre_Red-1.04305)	Biased	-58.48909	1726.629	-0.03	0.9730
(Rob_Blue-6.95869)*(Stre_Blue-1.04305)	Biased	-0.060216	999.6544	-0.00	1.0000
(Stab_Red-5.87047)*(Stab_Blue-5.87047)	Biased	0.0018842	0.001072	1.76	0.0787
(Stab_Red-5.87047)*(Stre_Red-1.04305)	Biased	0.7115681	0.062089	11.46	<.0001*
(Stab_Red-5.87047)*(Stre_Blue-1.04305)	Biased	-0.12349	0.056545	-2.18	0.0290*
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	Biased	-0.757498	0.058834	-12.88	<.0001*
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	Biased	0.2776047	0.059803	4.64	<.0001*
(Stre_Red-1.04305)*(Stre_Blue-1.04305)	Biased	-106.8427	3.41872	-31.25	<.0001*
(Conn_Red-6.49002)*(Conn_Red-6.49002)	Zeroed	0	0	.	.
(Conn_Blue-6.49002)*(Conn_Blue-6.49002)	Zeroed	0	0	.	.
(Disp_Red-6.912)*(Disp_Red-6.912)	Zeroed	0	0	.	.
(Disp_Blue-6.912)*(Disp_Blue-6.912)	Zeroed	0	0	.	.
(Pow_Red-9.3553)*(Pow_Red-9.3553)		1.8446759	0.063652	28.98	<.0001*
(Pow_Blue-9.3553)*(Pow_Blue-9.3553)		0.3851795	0.075441	5.11	<.0001*
(Rob_Red-6.95869)*(Rob_Red-6.95869)		0.9720857	42.62004	0.02	0.9818
(Rob_Blue-6.95869)*(Rob_Blue-6.95869)		0.0634749	38.47175	0.00	0.9987
(Stab_Red-5.87047)*(Stab_Red-5.87047)		0.0027407	0.000774	3.54	0.0004*
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)		0.0015217	0.000735	2.07	0.0384*
(Stre_Red-1.04305)*(Stre_Red-1.04305)		79.494735	2.36813	33.57	<.0001*
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)		37.132392	2.002664	18.54	<.0001*

Non-Linear Regression with All Metrics - After Removing Nonsignificant Terms and Collinear Relationships

Summary of Fit

RSquare	0.574954
RSquare Adj	0.574949
Root Mean Square Error	1.60647
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	36	13000896	361136	139934.7
Error	3.7e+6	9611183	2.580746	Prob > F
C. Total	3.7e+6	22612079		<.0001*

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	3.5e+6	9133342.5	2.58115	1.0031
Pure Error	185705	477840.1	2.57311	Prob > F
Total Error	3.7e+6	9611182.6		0.1777
				Max RSq
				0.9789

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	49.984559	0.014322	3490.0	<.0001*
Conn_Red	-0.098592	0.004127	-23.89	<.0001*
Conn_Blue	0.0549115	0.001116	49.20	<.0001*
Disp_Red	0.0188587	0.000689	27.38	<.0001*
Pow_Red	3.4210179	0.013483	253.72	<.0001*
Pow_Blue	-3.360196	0.014025	-239.6	<.0001*
Rob_Red	-0.091861	0.00968	-9.49	<.0001*
Stab_Red	0.1322134	0.000992	133.21	<.0001*
Stab_Blue	-0.119787	0.000978	-122.5	<.0001*
Stre_Red	-8.957721	0.078741	-113.8	<.0001*
Stre_Blue	9.116196	0.078595	115.99	<.0001*
(Conn_Red-6.49002)*(Conn_Blue-6.49002)	-0.000208	0.000142	-1.46	0.1430
(Disp_Red-6.912)*(Pow_Red-9.3553)	-0.281219	0.004014	-70.05	<.0001*
(Disp_Red-6.912)*(Pow_Blue-9.3553)	0.2909412	0.00392	74.21	<.0001*
(Disp_Red-6.912)*(Stab_Red-5.87047)	0.0083777	0.000436	19.23	<.0001*
(Disp_Red-6.912)*(Stab_Blue-5.87047)	-0.005098	0.000454	-11.22	<.0001*
(Disp_Red-6.912)*(Stre_Red-1.04305)	1.9883024	0.025881	76.82	<.0001*
(Disp_Red-6.912)*(Stre_Blue-1.04305)	-2.079654	0.025438	-81.75	<.0001*
(Pow_Red-9.3553)*(Pow_Blue-9.3553)	-2.874228	0.068117	-42.20	<.0001*
(Pow_Red-9.3553)*(Stab_Red-5.87047)	-0.116032	0.005206	-22.29	<.0001*
(Pow_Red-9.3553)*(Stab_Blue-5.87047)	0.1117379	0.00758	14.74	<.0001*
(Pow_Red-9.3553)*(Stre_Red-1.04305)	-30.80974	0.60794	-50.68	<.0001*
(Pow_Red-9.3553)*(Stre_Blue-1.04305)	19.637461	0.439681	44.66	<.0001*
(Pow_Blue-9.3553)*(Stab_Blue-5.87047)	-0.005765	0.006912	-0.83	0.4043
(Pow_Blue-9.3553)*(Stre_Red-1.04305)	23.880396	0.456428	52.32	<.0001*
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	-14.69071	0.487328	-30.15	<.0001*
(Stab_Red-5.87047)*(Stre_Red-1.04305)	1.1132445	0.043643	25.51	<.0001*
(Stab_Red-5.87047)*(Stre_Blue-1.04305)	-0.279953	0.008721	-32.10	<.0001*
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	-0.532264	0.049319	-10.79	<.0001*
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	-0.1778	0.048758	-3.65	0.0003*
(Stre_Red-1.04305)*(Stre_Blue-1.04305)	-163.0853	2.958061	-55.13	<.0001*
(Pow_Red-9.3553)*(Pow_Red-9.3553)	1.9013091	0.04571	41.60	<.0001*
(Pow_Blue-9.3553)*(Pow_Blue-9.3553)	1.0806937	0.034735	31.11	<.0001*
(Stab_Red-5.87047)*(Stab_Red-5.87047)	-0.00063	0.000512	-1.23	0.2185
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)	0.0015049	0.00053	2.84	0.0045*
(Stre_Red-1.04305)*(Stre_Red-1.04305)	104.42846	2.074766	50.33	<.0001*
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)	67.731796	1.779088	38.07	<.0001*

Non-Linear Regression with All Metrics – 2nd Iteration After Removing Nonsignificant Terms and Collinear Relationships

Summary of Fit

RSquare	0.574953
RSquare Adj	0.574949
Root Mean Square Error	1.60647
Mean of Response	50.00117
Observations (or Sum Wgts)	3724225

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	33	13000884	393966	152655.9
Error	3.7e+6	9611195	2.580747	**Prob > F**
C. Total	3.7e+6	22612079		<.0001*

Lack Of Fit

Source	DF	Sum of Squares	Mean Square	F Ratio
				1.0031
Lack Of Fit	3.5e+6	9133355.0	2.58115	**Prob > F**
Pure Error	185705	477840.1	2.57311	0.1776
Total Error	3.7e+6	9611195.1		**Max RSq**
				0.9789

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	49.982579	0.01416	3529.7	<.0001*
Conn_Red	-0.101086	0.003889	-25.99	<.0001*
Conn_Blue	0.0553525	0.001083	51.12	<.0001*
Disp_Red	0.0188309	0.000682	27.59	<.0001*
Pow_Red	3.420129	0.013271	257.71	<.0001*
Pow_Blue	-3.354216	0.013736	-244.2	<.0001*
Rob_Red	-0.097545	0.009145	-10.67	<.0001*
Stab_Red	0.132329	0.00099	133.70	<.0001*
Stab_Blue	-0.1197	0.000967	-123.8	<.0001*
Stre_Red	-8.933737	0.077103	-115.9	<.0001*
Stre_Blue	9.0969975	0.077009	118.13	<.0001*
(Disp_Red-6.912)*(Pow_Red-9.3553)	-0.282599	0.003803	-74.31	<.0001*
(Disp_Red-6.912)*(Pow_Blue-9.3553)	0.2917853	0.003848	75.82	<.0001*
(Disp_Red-6.912)*(Stab_Red-5.87047)	0.00841	0.000424	19.82	<.0001*
(Disp_Red-6.912)*(Stab_Blue-5.87047)	-0.005083	0.000419	-12.13	<.0001*
(Disp_Red-6.912)*(Stre_Red-1.04305)	1.9987477	0.024609	81.22	<.0001*
(Disp_Red-6.912)*(Stre_Blue-1.04305)	-2.085017	0.025204	-82.73	<.0001*
(Pow_Red-9.3553)*(Pow_Blue-9.3553)	-2.889523	0.063243	-45.69	<.0001*
(Pow_Red-9.3553)*(Stab_Red-5.87047)	-0.120722	0.003876	-31.15	<.0001*
(Pow_Red-9.3553)*(Stab_Blue-5.87047)	0.1093094	0.005107	21.40	<.0001*
(Pow_Red-9.3553)*(Stre_Red-1.04305)	-31.11121	0.541698	-57.43	<.0001*
(Pow_Red-9.3553)*(Stre_Blue-1.04305)	19.703056	0.425805	46.27	<.0001*
(Pow_Blue-9.3553)*(Stre_Red-1.04305)	23.959387	0.428817	55.87	<.0001*
(Pow_Blue-9.3553)*(Stre_Blue-1.04305)	-14.54969	0.478001	-30.44	<.0001*
(Stab_Red-5.87047)*(Stre_Red-1.04305)	1.1560244	0.029479	39.22	<.0001*
(Stab_Red-5.87047)*(Stre_Blue-1.04305)	-0.281325	0.008463	-33.24	<.0001*
(Stab_Blue-5.87047)*(Stre_Red-1.04305)	-0.517128	0.034306	-15.07	<.0001*
(Stab_Blue-5.87047)*(Stre_Blue-1.04305)	-0.222036	0.01233	-18.01	<.0001*
(Stre_Red-1.04305)*(Stre_Blue-1.04305)	-163.5261	2.884552	-56.69	<.0001*
(Pow_Red-9.3553)*(Pow_Red-9.3553)	1.9223445	0.040241	47.77	<.0001*
(Pow_Blue-9.3553)*(Pow_Blue-9.3553)	1.0757482	0.034577	31.11	<.0001*
(Stab_Blue-5.87047)*(Stab_Blue-5.87047)	0.0011172	0.000471	2.37	0.0178*
(Stre_Red-1.04305)*(Stre_Red-1.04305)	105.6292	1.863004	56.70	<.0001*
(Stre_Blue-1.04305)*(Stre_Blue-1.04305)	67.041536	1.697017	39.51	<.0001*

VITA

Mahmoud Turki Khasawneh obtained his B.S. degree in Management Information Systems from the Faculty of Economics and Administrative Sciences at the Hashemite University, in Zarqa, Jordan, in 2007. He received his Master of Engineering degree in Systems Engineering from the Department of Engineering Management and Systems Engineering at Old Dominion University, in 2009.

His research is centered on using modeling and simulation as a decision-making framework in business, operations management, information systems, and other fields. His most recent research focused on Network Centric Operations (NCO). NCO has an objective of creating a sense of shared situational awareness in an organization which will enable it to reap the benefits of high levels of connectivity to achieve a competitive advantage. Specifically, his dissertation focuses on the military aspect of distributed networked operations, for which he developed a discrete event simulation of a theoretical model of NCO called the Information Age Combat Model (IACM). Another research he has a keen interest in is modeling and simulation in healthcare. His interest in this field is focused on using modeling and simulation to aid decision-makers towards achieving cost reductions. He specifically intends to explore the utility of reform policies, such as a single-payer universal healthcare system, providing government tax breaks, a public health insurance option, and/or the expansion of information technology-supported patient record-keeping, and evaluate them as potential remedies to the troubled healthcare sector in the United States.