ABSTRACT

MEARS, CURTIS MICHAEL. Global Sensor Management: Enumeration of All Possible Allocations of Military Surveillance Assets. (Under the direction of Professor Thom J. Hodgson).

The United States has been interested in tracking the activities of its enemies, natural events, and even allies since before the Cold War, to protect its people from threats. To this end, the Department of Defense has tasked the Strategic Command (STRATCOM) with the daunting task of managing a majority of the surveillance assets used in this mission. STRATCOM needs to be sure that the probability of successfully protecting the United States and its allies is held to a high standard.

Success depends on assets' abilities to accomplish each function successfully when an event occurs. These assets can have the ability to perform multiple functions. The question of concern here is how to assign the assets with multiple abilities to optimize the probability of success.

This Thesis explains how to calculate the probabilities for a given allocation. Then it explains how a Visual Basic program enumerates each possible allocation, the inputs, options, and outputs of the program. This would allow a decision maker to pick the level of probability desired in each task.

Global Sensor Management: Enumeration of All Possible Allocations of Military Surveillance Assets

by
Curtis Michael Mears

A thesis submitted to the Graduate Faculty of
North Carolina State University
In partial fulfillment of the
Requirements for the degree of
Master of Science

Operations Research

Raleigh, North Carolina

2009

APPROVED BY:

_____	_____
Thom J. Hodgson Committee Chair	Russell E. King
_____	
Michael G. Kay	

DEDICATION

To my wife,
Michelle.

For the love and support, she has given and will continue to give. I truly believe that behind most strong and successful men is a stronger and more successful woman.

Thank-you

BIOGRAPHY

Curtis Michael Mears was born in Dayton, Ohio in 1979. He graduated from Tri-County North High School in 1999 and went on to Wittenberg University where he graduated in 2003 with a Bachelor of Arts in Mathematics. He started his Operations Research Career as a Department of Army Intern with the United States Army Special Operations Command (USASOC). Upon completion of the intern program, USASOC offered him a full time position testing equipment in their Test and Evaluation Division. After just two short years, the Army granted him the opportunity to return to school fulltime to complete a Masters Degree in Operations Research. Upon completion of this degree, he will return to his former position within USASOC.

Curtis is married to Michelle. They have a son, Tobias who is two years old, and a daughter, Lucia who was born just after the fall semester in 2007.

ACKNOWLEDGMENTS

I would like to recognize and thank the following people:

The Inner Circle – You know who you are and what you did; Kristin, John, Thom, and a special thanks to Ben. This would not have been possible with your help.

Mr. Uwe Stutz – For mentoring me into my career and your continuing support and friendship. A special thanks for taking on extra work while I attended school.

My Chain of Command and those that supported me when I applied for this opportunity especially:

Mr. James B. Taylor
COL Todd L. Dodson
COL Jeffery Putz
LTC Joseph A. Capobianco
Mr. Michael J. Collins
LTC Christian R. Larlee
Mr. Michael G. McCaskill
Ms. Doris A. Taylor
Ms. Martha Newman

My parents, Michael and Christy Mears – For putting aside our friendship and raising me to be a successful human being. I can only hope that I am half as successful as you were when it comes to raising a child.

For guidance that had an impact on my life more than they know, Mr. Ronald A. Spencer, Dr. Douglas M. Andrews, Dr. John Whitaker, and, finally, Mr. David Unger. This is probably my largest apple and I'm still hungry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Background

The United States has been interested in tracking the activities of its enemies, natural events, and even allies since before the Cold War, to protect its people from threats. To this end, the Department of Defense has tasked the Strategic Command (STRATCOM) with the daunting task of managing a majority of the surveillance assets used in this mission. STRATCOM needs to be sure that the probability of successfully protecting the United States and its allies is held to a high standard. STRATCOM has many assets that it can employ to accomplish this mission that they have broken up into four competing tasks. These tasks are Missile Defense, Space Surveillance, Intelligence Collection, and Missile Warning. These are competing tasks in that they all need to be accomplished. Many of the assets that assist in these tasks cannot work on more than one of them at a time. This research is a continuation of the research preformed by Arney [Arney, 2008] and has been use by Dulin to test a heuristic to find the optimal solution [Dulin, 2009].

1.2 The Problem

The four tasks that STRATCOM is concerned with are Missile Defense, Space Surveillance, Intelligence Collection, and Missile Warning. Assigned to each task is an event type to handle. For the Missile Defense, Task 1, a missile launched at the US or an ally is an event. Each of these tasks consists of a series of functions. The Missile Defense Task consists of nine functions (STRATCOM starts by numbering the first function, Function 2). Function 2 is Detect, then Cue, Track, Classify, Engage, Track Update, Shoot, Kill, and finally, Function 10, Kill Assess. Successful completion of each function in sequence is required for total task success. Assets are assigned to perform each function. Usually multiple assets are assigned to perform the same function in parallel and many assets can perform multiple functions in a task. However, assets cannot always perform functions in multiple tasks. Deciding which task an asset should support is the problem. These assets then pass information to assets that

perform the next function. STRATCOM numbered these assets M1 through M36. In most places within this thesis, the "M" has been dropped for simplicity.

STRATCOM has provided the example in Figure 1.2 through Figure 1.5 to help understand the problem. The nodes represent an asset and bare the name of the asset. An arc represents the ability to pass information from an asset assigned to the previous function to an asset assigned to perform the next function. At this time Tasks 2, 3, and 4 were not developed. For the purpose of this study, a simple three function, asset only passing information from and to itself structure was used, Figure 1.3 through Figure 1.5. Further note, that STRATCOM skips Function 3 in their numbering scheme. This simple structure is referred to as linear in this thesis. Figure 1.1 is a legend to the color-coding used in the STRATCOM example.

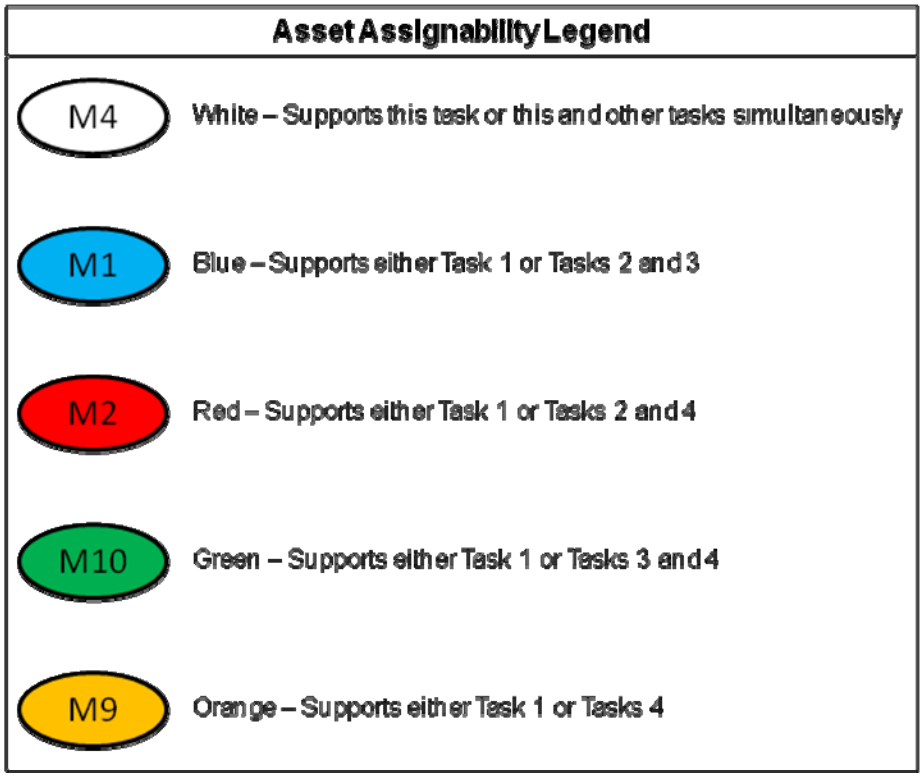


Figure 1.1: Asset Assignability Legend

If an asset is white, then it only has the ability to work on the task that it appears or it has the ability to work on the multiple tasks in which it appears simultaneously; either way, there is no need to consider a white asset for any task assignment decisions. If an asset is blue, either it is dedicated to Task 1 or it supports both Tasks 2 and 3 simultaneously. If an asset is red, either it is dedicated to Task 1 or it supports both Tasks 2 and 3 simultaneously. If an asset is green, either it is dedicated to Task 1 or it supports both Tasks 3 and 4. If an asset is orange, it is dedicated to either Task 1 or Task 4.

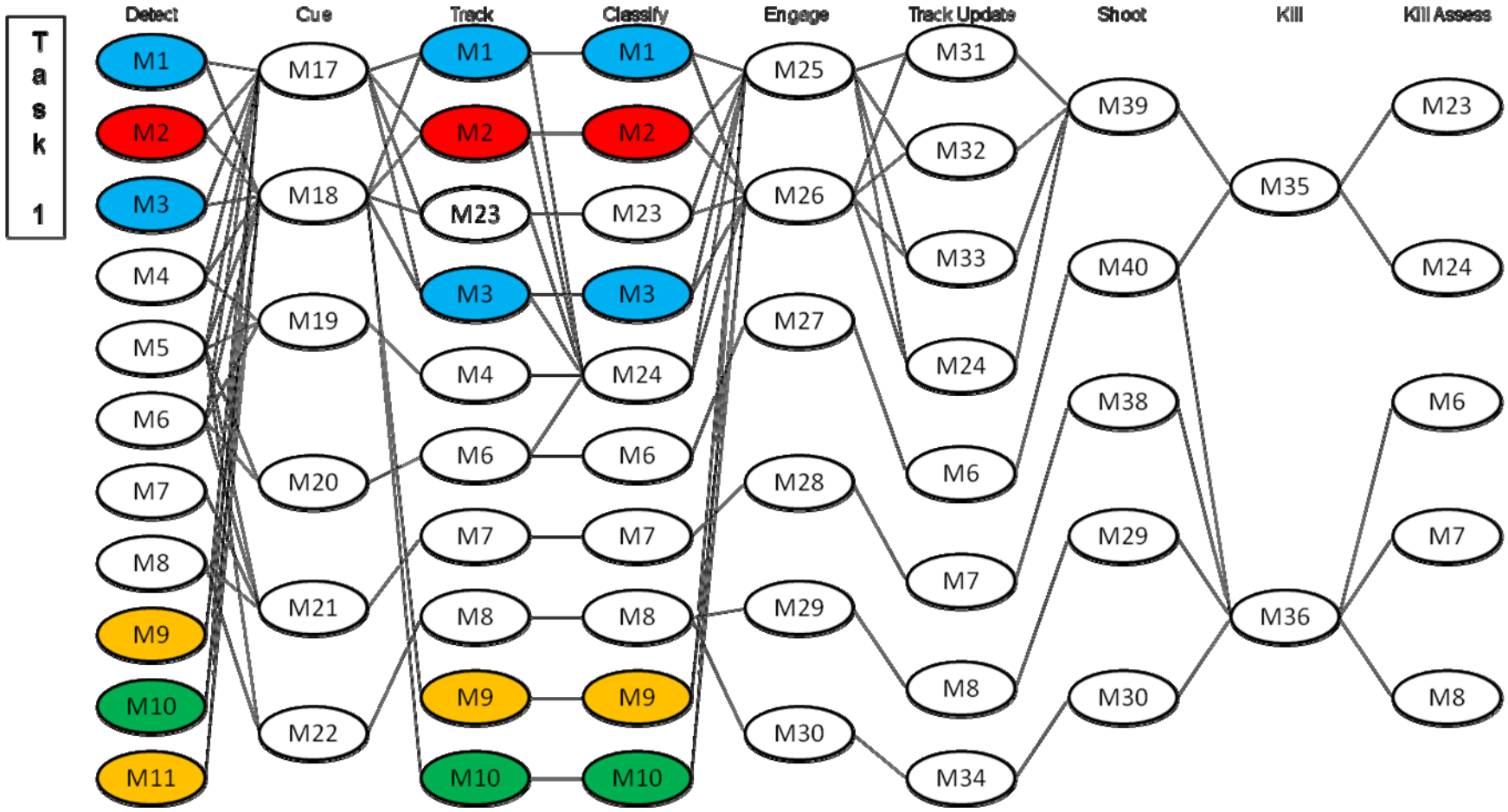


Figure 1.2: Task 1 Assets and Connections

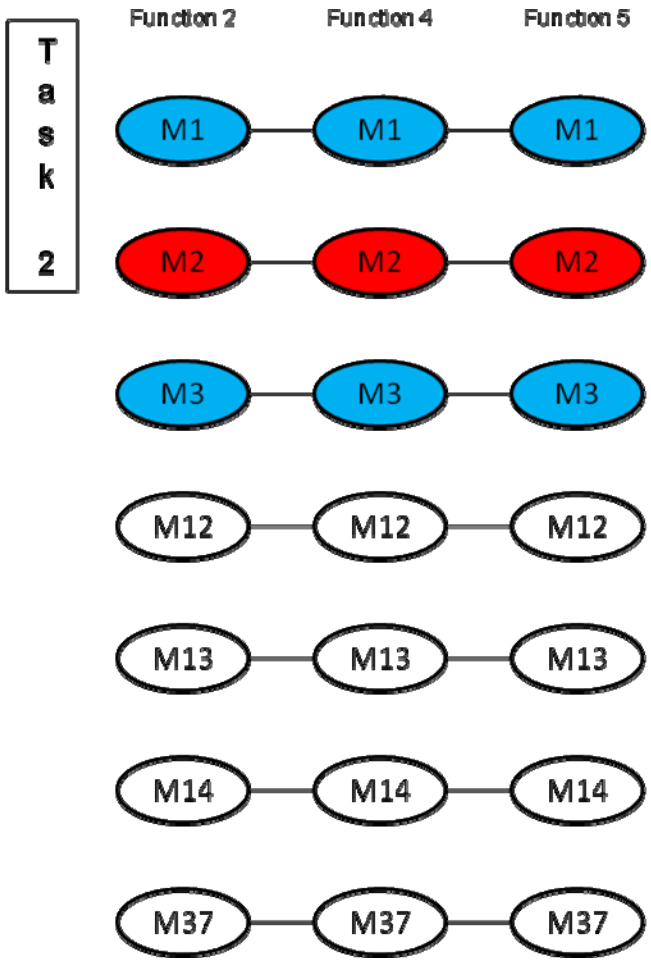


Figure 1.3: Task 2 Assets and Connections

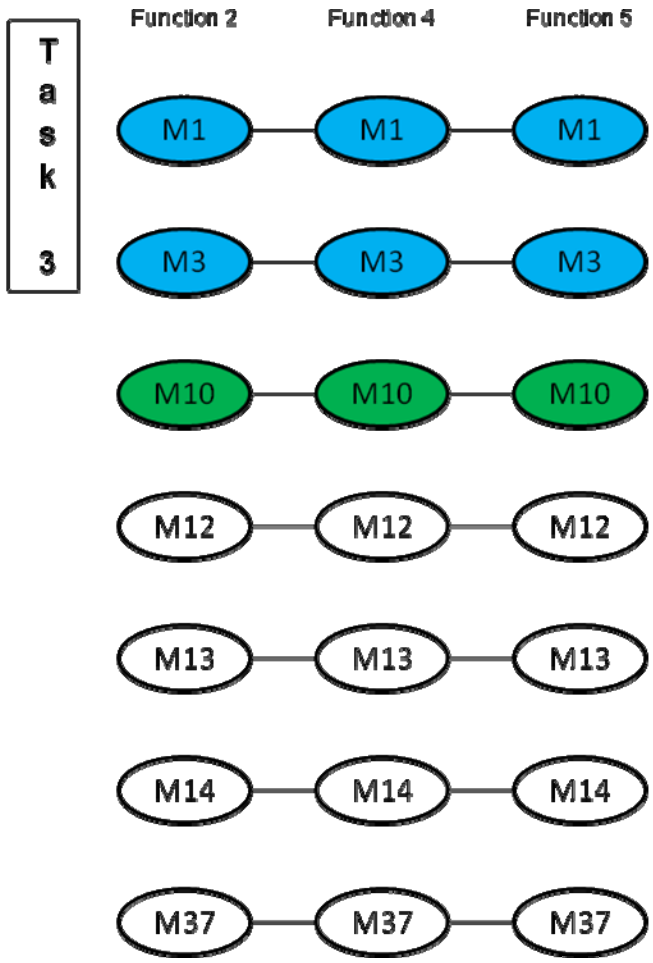


Figure 1.4: Task 3 Assets and Connections

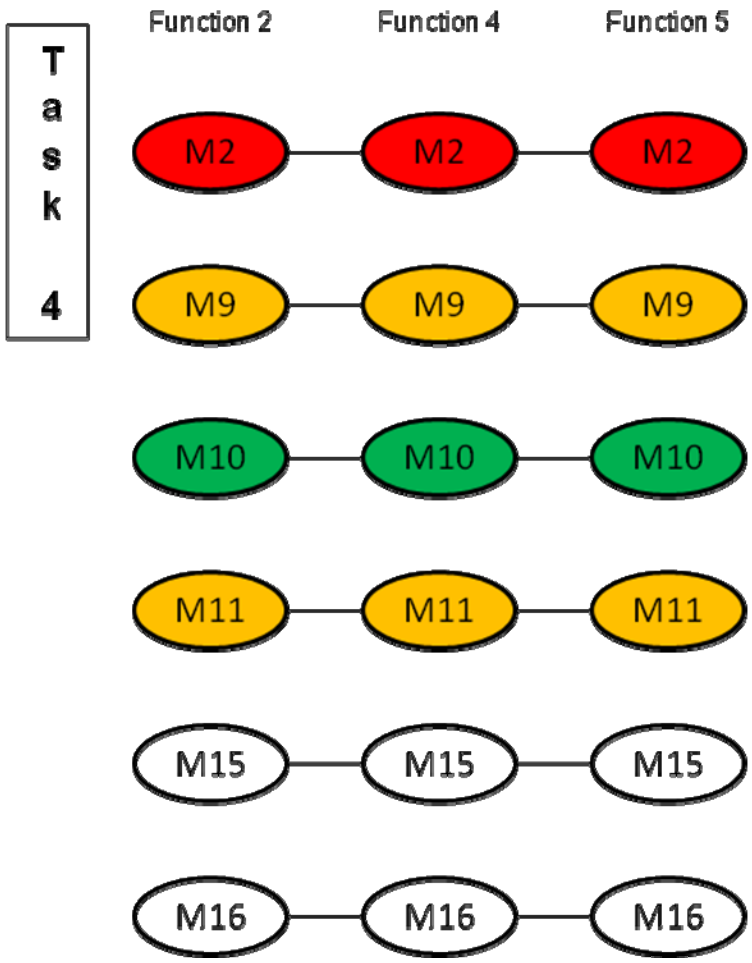


Figure 1.5: Task 4 Assets and Connections

The question of concern is how to assign the colored assets to perform the functions within the tasks to optimize the probability of success for Task 1 while maintaining a specified level of probability of success of the of tasks.

1.3 Key Assumptions

The following assumptions simplified the problem for study.

1.3.1 Tasks 2, 3, and 4 are linear

This assumption is the result of the undeveloped Tasks 2, 3, and 4 presented by SRATCOM. This assumption simplifies the calculation of the probability of success of these tasks and drives the assumption outlined in section 1.3.3. Outlined in section 2.6.1 is how to analyze a network with more than one complicated task.

1.3.2 Static Problem

The problem is static. In the real problem, an asset's availability would most likely change over time. This means that a solution to this static problem would only be valid for a specific period.

1.3.3 All decisions are made in Function 2

This assumption follows from the assumption in section 1.3.1. Since Tasks 2, 3, and 4 are linear, it would be a waste to assign an asset to one and then reassign it before it completes the task. Outlined in section 2.6.2 is how to analyze a network with the ability to make decisions in later functions.

1.3.4 Single Event

It is unclear from the STRATCOM example if assets can handle one or multiple events simultaneously. It is the case that some of the assets have an upper limit no the number of events they can handle and therefore would be unavailable if there were multiple events.

1.4 Thesis Overview

Explained in the next chapter is the methodology to calculate the probabilities for a given allocation. Then Chapter 2 details an enumeration scheme for each possible allocation based on the inputs, options, and outputs of the program. This allows a decision maker to select the level of the probability of success desired in each task. This type of enumeration takes time when time may not available in a real world situation. In Chapter 3, the STRATCOM example is enumerated and analyzed. Finally, Chapter 4 outlines possible future work in this area.

2 Enumerating the Problem

2.1 Solving a Network by Hand

Using the simple network found in Figure 2.1 to examine the mathematics to calculate the probability of success, consider Asset *A* assigned to Task1, Function 2. Let the probability it successfully sends correct information to an asset in Function 3 be $P(A_{1,2})$. For assets assigned to subsequent functions, like Asset *F* assigned to Task 1, Function 3. Let the probability it successfully sends correct information to an asset in Function 4 given that it received correct information from only Asset *A* in Function 2 be $P(F_{1,3}| A_{1,2})$. Furthermore, let the probability that Asset *H*, assigned to Task 1, Function 3 successfully sends correct information to an asset in the Function 4 given that it received correct information from only Assets *A* and *B* in Function 2 be $P(H_{1,3}| A_{1,2}, B_{1,2})$ and if Asset *H* also successfully received information from Asset *C* this is denoted as $P(H_{1,3}| A_{1,2}, B_{1,2}, C_{1,2})$. Note that this notation leaves out the failure of an asset in the previous function. This will simplify notation, but the failure of an asset is important part of the calculation of the probability of success of a network.

In using the Law of Total Probability to calculate the probability of success for a task, STRACOM would need to know the probability of success of an asset when it successfully receives information from all possible combinations of the assets that can send information to it [Wackerly, et al, 2002] and [Walpole, et al, 2002]. In most cases, STRATCOM has not done this research or the combining of the information is not possible. In these cases, STRATCOM uses the asset's information that maximizes the probability of success [Arney, 2008]. So if $P(H_{1,3}| A_{1,2}) > P(H_{1,3}| B_{1,2}) > P(H_{1,3}| C_{1,2})$, then when Asset *H* successfully receives information from Assets *A* and *B*, Asset *H* will only use the information from Asset *A* and $P(H_{1,3}| A_{1,2}, B_{1,2}) = P(H_{1,3}| A_{1,2})$. Additionally, $P(H_{1,3}| A_{1,2}, C_{1,2}) = P(H_{1,3}| A_{1,2})$, $P(H_{1,3}| B_{1,2}, C_{1,2}) = P(H_{1,3}| B_{1,2})$, and $P(H_{1,3}| A_{1,2}, B_{1,2}, C_{1,2}) = P(H_{1,3}| A_{1,2})$. This greatly limits the amount of data needed to calculate the probability for a network and if, in fact, some

assets do combine information then the calculated probability would serve as a lower bound on the true probability of success.

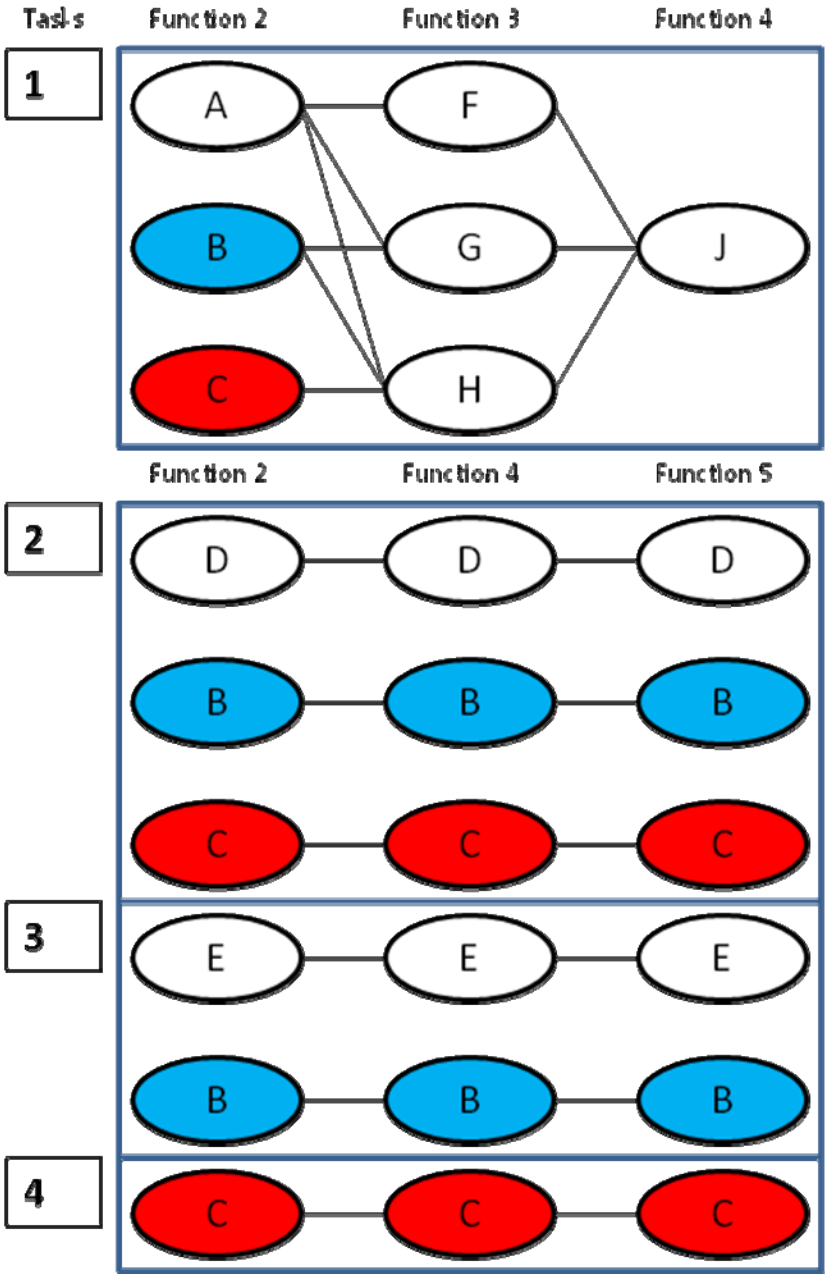


Figure 2.1: Simple Four Task Network

Using the example in Figure 2.1, if Assets *B* and *C* are both assigned to Task 1, to calculate of the probability of success for each of the tasks, $P(A_{1,2})$, $P(B_{1,2})$, $P(C_{1,2})$, $P(D_{2,2})$, $P(E_{3,2})$, $P(F_{1,3}|\ A_{1,2})$, $P(G_{1,3}|\ A_{1,2})$, $P(G_{1,3}|\ B_{1,2})$, $P(H_{1,3}|\ A_{1,2})$, $P(H_{1,3}|\ B_{1,2})$, $P(H_{1,3}|\ C_{1,2})$, $P(D_{2,4}|\ D_{2,2})$, $P(E_{3,4}|\ E_{3,2})$, $P(J_{1,4}|\ F_{1,3})$, $P(J_{1,4}|\ G_{1,3})$, $P(J_{1,4}|\ H_{1,3})$, $P(D_{2,5}|\ D_{2,4})$, and $P(E_{3,5}|\ E_{3,4})$ need to be known. The $P(B_{2,2})$, $P(C_{2,2})$, $P(B_{3,2})$, $P(C_{4,2})$, $P(B_{2,4}|\ B_{2,2})$, $P(C_{2,4}|\ C_{2,2})$, $P(B_{3,4}|\ B_{3,2})$, $P(C_{4,4}|\ C_{4,2})$, $P(B_{2,5}|\ B_{2,4})$, $P(C_{2,5}|\ C_{2,4})$, $P(B_{3,5}|\ B_{3,4})$, and $P(C_{4,5}|\ C_{4,4})$ are all set to zero. Finally, let $P(G_{1,3}|\ A_{1,2}) > P(G_{1,3}|\ B_{1,2})$, $P(H_{1,3}|\ A_{1,2}) > P(H_{1,3}|\ B_{1,2}) > P(H_{1,3}|\ C_{1,2})$, and $P(J_{1,4}|\ F_{1,3}) > P(J_{1,4}|\ G_{1,3}) > P(J_{1,4}|\ H_{1,3})$ so $P(G_{1,3}|\ A_{1,2}, B_{1,2}) = P(G_{1,3}|\ A_{1,2})$, $P(H_{1,3}|\ A_{1,2}, B_{1,2}) = P(H_{1,3}|\ A_{1,2}, C_{1,2}) = P(H_{1,3}|\ A_{1,2}, B_{1,2}, C_{1,2}) = P(H_{1,3}|\ A_{1,2})$, $P(H_{1,3}|\ B_{1,2}, C_{1,2}) = P(H_{1,3}|\ B_{1,2})$, $P(J_{1,4}|\ F_{1,3}, G_{1,3}) = P(J_{1,4}|\ F_{1,3}, H_{1,3}) = P(J_{1,4}|\ F_{1,3}, G_{1,3}, C_{1,2}) = P(J_{1,4}|\ F_{1,3})$, and $P(J_{1,4}|\ G_{1,3}, H_{1,3}) = P(J_{1,4}|\ G_{1,3})$. Using an adapted Law of Total probability the calculation for this network is as follows [Wackerly, et al, 2002] and [Walpole, et al, 2002]:

Task 1, Function 3

$$P(A_{1,3}) = P(A_{1,2})\ P(F_{1,3}|\ A_{1,2})$$

$$P(G_{1,3}) = P(A_{1,2})\ (1 - P(B_{1,2}))\ P(G_{1,3}|\ A_{1,2}) + (1 - P(A_{1,2}))\ P(B_{1,2})\ P(G_{1,3}|\ B_{1,2}) + P(A_{1,2})\ P(B_{1,2})\ P(G_{1,3}|\ A_{1,2})$$

$$\begin{aligned} P(H_{1,3}) = {} & P(A_{1,2})\ (1 - P(B_{1,2}))\ (1 - P(C_{1,2}))\ P(H_{1,3}|\ A_{1,2}) \\ & + (1 - P(A_{1,2}))\ P(B_{1,2})\ (1 - P(C_{1,2})\ P(H_{1,3}|\ B_{1,2}) \\ & + (1 - P(A_{1,2}))\ (1 - P(B_{1,2}))\ P(C_{1,2})\ P(H_{1,3}|\ C_{1,2}) \\ & + P(A_{1,2})\ P(B_{1,2})\ (1 - P(C_{1,2}))\ P(H_{1,3}|\ A_{1,2}) \\ & + P(A_{1,2})\ (1 - P(B_{1,2}))\ P(C_{1,2})\ P(H_{1,3}|\ A_{1,2}) \\ & + (1 - P(A_{1,2}))\ P(B_{1,2})\ P(C_{1,2})\ P(H_{1,3}|\ B_{1,2}) \\ & + P(A_{1,2})\ P(B_{1,2})\ P(C_{1,2})\ P(H_{1,3}|\ A_{1,2}) \end{aligned}$$

Task 1, Function 4

$$P(J_{1,4}) = P(F_{1,3})\,(1 - P(G_{1,3}))\,(1 - P(H_{1,3}))\,P(J_{1,4}|\,F_{1,3})$$
$$+ (1 - P(F_{1,3}))\,P(G_{1,3})\,(1 - P(H_{1,3})\,P(J_{1,4}|\,G_{1,3})$$
$$+ (1 - P(F_{1,3}))\,(1 - P(G_{1,3}))\,P(H_{1,3})\,P(J_{1,4}|\,H_{1,3})$$
$$+ P(F_{1,3})\,P(G_{1,3})\,(1 - P(H_{1,3}))\,P(J_{1,4}|\,F_{1,3})$$
$$+ P(F_{1,3})\,(1 - P(G_{1,3}))\,P(H_{1,3})\,P(J_{1,4}|\,F_{1,3})$$
$$+ (1 - P(F_{1,3}))\,P(G_{1,3})\,P(H_{1,3})\,P(J_{1,4}|\,G_{1,3})$$
$$+ P(F_{1,3})\,P(G_{1,3})\,P(H_{1,3})\,P(J_{1,4}|\,F_{1,3})$$

Task 2

$$P(D_{2,5}) = P(D_{2,2})\,P(D_{2,4}|\,D_{2,2})\,P(D_{2,5}|\,D_{2,4})$$

Task 3

$$P(E_{2,5}) = P(E_{2,2})\,P(E_{2,4}|\,E_{2,2})\,P(E_{2,5}|\,E_{2,4})$$

Task 4 has no assets assigned to it.

This is only one of the three ways to allocate Assets *B* and *C*. In order to demonstrate these calculations, consider assigning the probabilities in Table 2.1 to the current example in Figure 2.1.

Table 2.1: Probabilities for an Example

Known Probabilities								
Function 2			Function 3			Function 4		
Task 1								
$P(A_{1,2})$	0.99		$P(F_{1,3}\mid A_{1,2})$	0.59		$P(J_{1,4}\mid F_{1,3})$	0.87	
$P(B_{1,2})$	0.94		$P(G_{1,3}\mid A_{1,2})$	0.77		$P(J_{1,4}\mid G_{1,3})$	0.67	
$P(C_{1,2})$	0.66		$P(G_{1,3}\mid B_{1,2})$	0.56		$P(J_{1,4}\mid H_{1,3})$	0.53	
			$P(H_{1,3}\mid A_{1,2})$	0.98				
			$P(H_{1,3}\mid B_{1,2})$	0.80				
			$P(H_{1,3}\mid C_{1,2})$	0.78				
Function 2			Function 4			Function 5		
Task 2								
$P(D_{2,2})$	0.62		$P(D_{2,4}\mid D_{2,2})$	0.85		$P(D_{2,5}\mid D_{2,4})$	0.68	
$P(B_{2,2})$	0.72		$P(B_{2,4}\mid B_{2,2})$	0.63		$P(B_{2,5}\mid B_{2,4})$	0.74	
$P(C_{2,2})$	0.84		$P(C_{2,4}\mid C_{2,2})$	0.59		$P(B_{2,5}\mid B_{2,4})$	0.73	
Task 3								
$P(E_{3,2})$	0.86		$P(E_{3,4}\mid E_{3,2})$	0.65		$P(E_{3,5}\mid E_{3,4})$	0.88	
$P(B_{3,2})$	0.55		$P(B_{3,4}\mid B_{3,2})$	0.60		$P(B_{3,5}\mid B_{3,4})$	0.69	
Task 4								
$P(C_{4,2})$	0.57		$P(C_{4,4}\mid C_{4,2})$	0.90		$P(C_{4,5}\mid C_{4,4})$	0.95	

STRATCOM cannot disclose the exact details of the probabilities for obvious reasons. The probabilities used here are solely for this demonstration. See Chapter 3 for more details.

Task 1, Function 2 is given.

Task 1, Function 3

$P(F_{1,3}) = 0.99 * 0.59 = 0.5841$

Table 2.2: Probability of Success of Asset *G*

Asset *G*			
Ways to Involve Asset *G*	Probability of this Success	Given Probability of Success of Asset *G*	Probability of the Success of Asset *G* and its Predecessors
A, B	0.9504	0.77	0.7318
A, < >*B*	0.0594	0.77	0.0457
< >*A, B*	0.0094	0.56	0.0053
< >*A, B*	0.0004	0.00	0.0000
Probability of Success for Asset *G*			$P(G_{1,2}) = 0.7828$

Let < >*x* indicate the failure of *x*.

Table 2.3: Probability of Success of Asset *H*

Asset *H*			
Ways to Involve Asset *H*	Probability of this Success	Given Probability of Success of Asset *H*	Probability of the Success of Asset *H* and its Predecessors
A, B, C	0.6142	0.98	0.6019
A, B, < >*C*	0.3164	0.98	0.3101
A, < >*B, C*	0.0392	0.98	0.0384
< >*A, B, C*	0.0062	0.80	0.0050
A, < >*B,* < >*C*	0.0202	0.98	0.0198
< >*A, B,* < >*C*	0.0032	0.80	0.0026
< >*A,* < >*B, C*	0.0004	0.78	0.0003
< >*A,* < >*B,* < >*C*	0.0002	0.00	0.0000
Probability of Success for Asset *H*			$P(H_{1,2}) = 0.9781$

Let < >*x* indicate the failure of *x*.

Task 1, Function 4

Table 2.4: Probability of Success of Asset *J*

Asset *J*			
Ways to Involve Asset *J*	Probability of this Success	Given Probability of Success of Asset *J*	Probability of the Success of Asset *J* and its Predecessors
F, *G*, *H*	0.4472	0.87	0.3891
F, *G*, < >*H*	0.0100	0.87	0.0087
F, < >*G*, *H*	0.1241	0.87	0.1080
< >*F*, *G*, *H*	0.3184	0.67	0.2133
F, < >*G*, < >*H*	0.0028	0.87	0.0024
< >*F*, *G*, < >*H*	0.0071	0.67	0.0048
< >*F*, < >*G*, *H*	0.0884	0.53	0.0469
< >*F*, < >*G*, < >*H*	0.0020	0.00	0.0000
Probability of Success for Asset *J*			$P(J_{1,3}) = 0.7732$

Let < >*x* indicate the failure of *x*.

Task 2

$P(D_{2,5}) = 0.62 * 0.85 * 0.68 = 0.3584$

Task 3

$P(E_{3,5}) = 0.86 * 0.65 * 0.88 = 0.4919$

Assigning no assets to perform Task 4, the probability of success is zero.

To summarize, if Assets *B* and *C* are both assigned to Task 1, the probability of successful completion of the tasks are 0.7732, 0.3584, 0.4919, 0.0000; respectively. Table 2.5 below summarizes all possible assignments of Assets *B* and *C*.

Table 2.5: Enumeration of Example

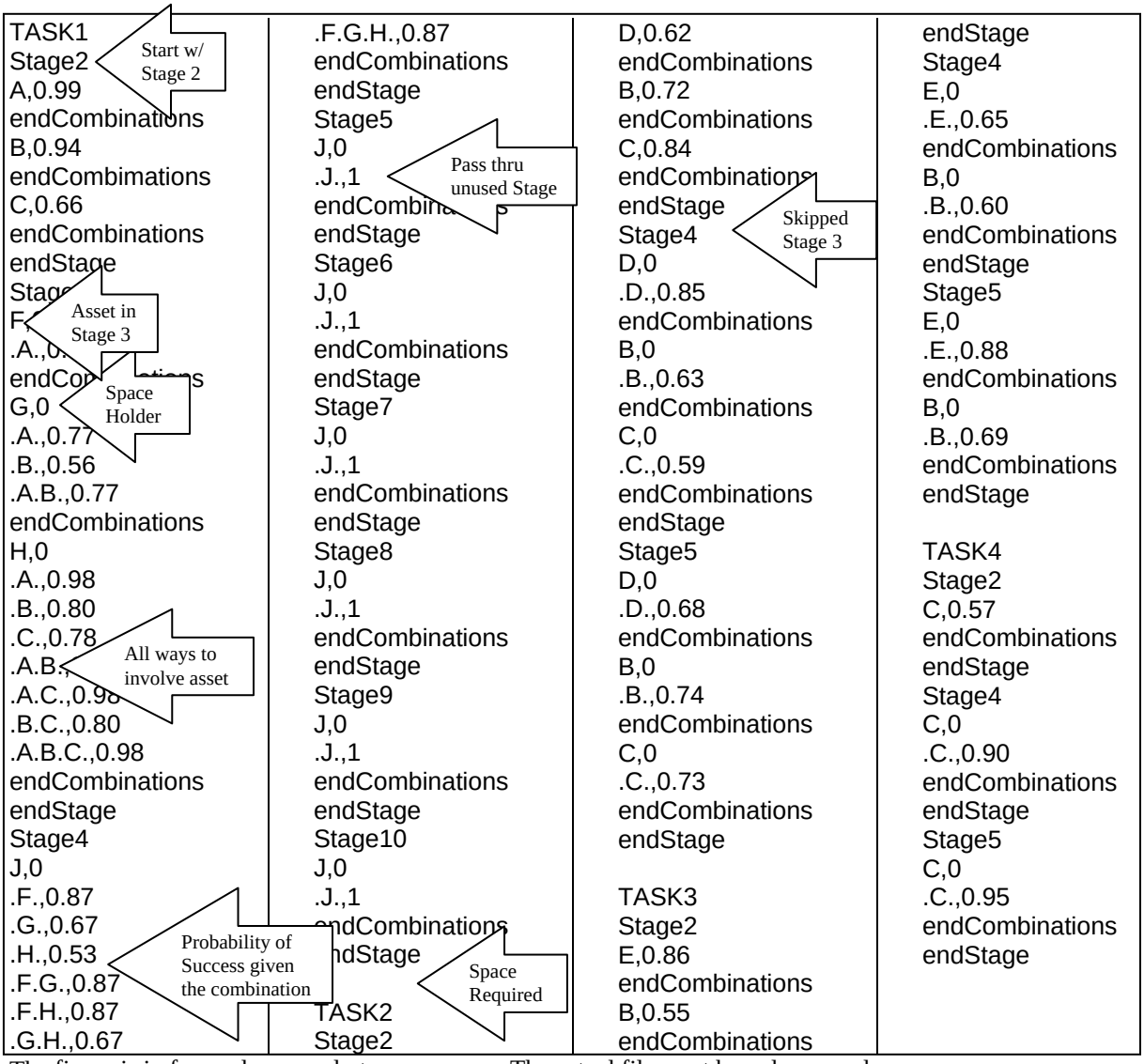
Assignable Assets Assigned to Task 1	Probability of Success			
	Task 1	Task 2	Task 3	Task 4
B, C	0.7732	0.3584	0.4919	0.0000
B	0.7721	0.5905	0.4919	0.4874
C	0.7717	0.5737	0.6076	0.0000
none	0.7714	0.7280	0.6076	0.4874

2.2 Computer Code Overview

To study the STRATCOM example, Figure 1.1 though Figure 1.5, a computer program was built in Visual Basic (VB), Appendix B. The code uses the VB feature of dictionaries to organize the assets and their connections. It calculates all the possible assignments and then calculates the probability of success of each assignment. The program can only handle a complex network in Task 1 and provides the probability of success for all of the assets assigned to the last function. For Tasks 2, 3, and 4, the program requires them to be linear and computes the total success of the task rather than the individual asset's probability of success assigned to the last function. The program has three different modes for different types of information. The following sections detail these differences. The program inputs are comma-delimited files with the extension "csv." Microsoft® Excel, WordPad, or Notepad will create these types of files. In addition, the output of the program saves in comma-delimited files. The term "function" as been replaced in the program with "stage" because the term "function" is a command in VB. The program is expecting that there are assets in all nine functions of Task 1 and the three functions of the others. The examples in Figure 2.2 and Figure 2.3 show how to shorten the number of functions to only three, but the program cannot handle more at this time. Remember that the STRATCOM example, Figure 1.1 through Figure 1.5, starts with Function 2 and Task 2, 3, and 4 skip Function 3, so the program does the same and the input must reflect this, too.

2.3 Calculating a Network with Known Probabilities

If the user knows the probabilities for a network, the program can generate the total probability of success for each task for each combination of assignable assets. This is option "e." The input file must be a comma delimited file formed as shown in Figure 2.2. When determining the probability of success of an asset receiving information from more than one asset does not have to be set to the maximum of the asset given information from only one of the assets. This allows for the combining of information and increasing the probability of success when appropriate. Figure 2.2 is of the three stage example in Figure 2.1, notice that all nine functions in Task 1 must have assets and how to have the program not change the result when the function is not needed. Note that all of the tasks start with Stage 2 and tasks 2 through 4 skip Stage 3, just as the STRATCOM example.



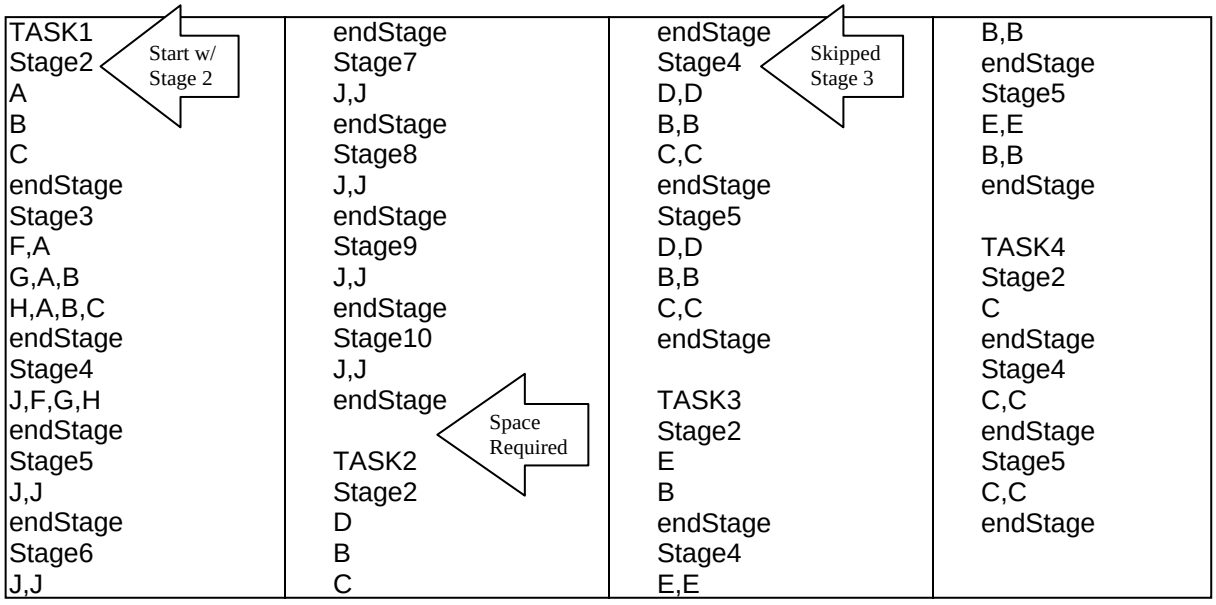
The figure is in four columns only to save space. The actual file must be only one column.

Figure 2.2: Input File for Option "e"

2.4 Calculating a Network with Random Probabilities

If the user does not know the probabilities of the network, the program can randomly assign probabilities to assist in analyzing the network. The program asks for a range in which it will sample the probabilities of success of an asset from a uniform density function. Currently, it asks for a range that the user wants the probabilities of the assets to fall in for the assets in

Function 2 of Task 1 and then a range for the other machines. If this ability to make different functions of Tasks sample from different ranges is helpful, others could be added. The study of the STRATCOM example presented in Chapter 3 does not use this feature, because there is no reason to change these assets to a different range. Furthermore, when an asset receives information from more than one asset in the previous stage it takes the highest probability of the single success as its probability of success, just as in the hand calculation in section 2.1. When specifying this range the program is expecting a percentage; so for 80% probability "80" is entered into the program not "0.8." Use option "n" to run one such network and option "r" for multiple runs. If option "r" is used the output will be the average probability of success for all the runs. The input file for either of these options is a comma delimited; Figure 2.3 is an example of the input file for the example in Figure 2.1. Note that the file start with Stage 2 and Tasks 2 through 4 skip Stage 3, again. In addition, all stages must be used. The location of the output changes because this example does not use all of the Stages in Task 1, detailed in Section 2.5.



The figure is in four columns only to save space. The actual file must be only one column.

Figure 2.3: Input File for Options "n" or "r"

2.5 Program Output

All output from the program is in comma-delimited files. One of the output files from the program is similar for all options. This is the final probability file. For option "r" this file holds the averages of all the runs rather than final probability of each combination of asset assigned to Task 1. Figure 2.4 is the output when the program runs the comma delimited file in Figure 2.2. The program will ask the user to name it and it will overwrite a file of that same name without warning. In the file's first column are all combinations of the machines assigned to Task 1 starting with none of them and ending with all of them. The next columns, excluding the last three, are the final assets in Task 1 and the corresponding probability of success for that asset in the final function given the combination of assignable assets. This is to allow the study of final probability of success of each asset rather than the total probability of success of the task. To calculate the total probability of success for this task the complement of all assets failing [Scheaffer, 1995] combined with rule of independent probabilities [Ross, 2007] to form the following equation:

$$P(\text{Task 1 Success}) = 1 - (1 - P(1_{1,10}))\,(1 - P(2_{1,10}))\,\ldots\,(1 - P(n_{1,10})),$$

For n assets in Function 10 of Task 1, where assets are numbered 1 to n.

The program performs this calculation on the other tasks and the result is report in the last three columns of the file.

```
NETWORK ALLOCATION,J,TASK2,TASK3,TASK4
No assets on task 1.,0.77141829396258,0.72795284835166,0.607609816,0.48735
.B.,0.772144661872568,0.59049765232,0.49192,0.48735
.C.,0.771688025985769,0.57373544896,0.607609816,0
.B.C.,0.772160487391931,0.35836,0.49192,0
```

Figure 2.4: Comma Delimited File Output from Program

This file is opened in Microsoft's® Excel will look like Figure 2.5 and the columns are more clear.

NETWORK ALLOCATION	J	TASK2	TASK3	TASK4
No assets on task 1.	0.771418	0.727953	0.60761	0.48735
.B.	0.772145	0.590498	0.49192	0.48735
.C.	0.771688	0.573735	0.60761	0
.B.C.	0.77216	0.35836	0.49192	0

Figure 2.5: Output from Program Opened in Microsoft's® Excel

In addition, for all options, comma delimited files named "Stg*n*" for *n* from 3 to 10 are produced. Within these files are the Task 1, Function *n* asset probabilities. These files are identical to the files shown in Figure 2.4 and Figure 2.5, except that the columns for Tasks 2 through 4 do not exist. For option "r," they will be the last iteration. Unless the user moves these files or changes their names they will be over wrote with each run of the program. In the example, Task 1 only uses Functions 2, 3, and 4 so the final probability file is incorrect for options "n" and "r" because all of the assets in Functions 5 through 10 receive random probabilities. Option "r" is not able to produce correct results and option "n" the correct end function probabilities are found in the file "Stg4" since Function 4 was the last function used in Task 1 of the example, Figure 2.1. In Figure 2.2 all of the unused assets are given one as the probability of success this passes the correct probability of success to the final stage for option "e." Option "n" has one additional output file that the other options do not have. This is an output of the random probabilities assigned to the assets. This file is formatted just as the input file for option "e," Figure 2.2. This allows a user to generate random probabilities for a given network, view them, change them, and rerun with option "e."

2.6 Using the Program to Challenge Some Assumptions

In Section 1.3 some assumptions were outlined that the program has been wrote to. Use the program in the following ways to challenge them.

2.6.1 Challenging the Assumption that Tasks 2, 3, and 4 are Linear

If both, Tasks 1 and 2 are complicated then the program needs ran once with Task 1 as it is and Task 2 only needs the assignable assets and a their linear connections. Save the output file. Then replace the Task 1 network with the actual Task 2 network and run the program again. The output from this run is the data for Task 2. In addition, if Task 2 does not have nine functions then the corresponding "Stg*n*" file contains the final probability of success for each asset in the final function of Task 2. Finally, match up the two output files, when an asset is assigned to Task 1 it is unavailable to Task 2. This scheme can be applied to as many complicated networks that share assignable assets by running them two at a time, ignoring the output for Tasks 2, 3, and 4 then matching up the assigned assets in the output files.

2.6.2 Challenging the Assumption that All Decisions Must be made in Function 2

To have the program simulate making decisions in later functions place such an asset in Function 2 and only connect it to itself in all subsequent functions until it is in the actual function that it should appear and give it the additional connections from the previous function that it actual has. Assign the probability of success of this asset given that it receives information from itself one until the final function that it receives information from itself, then its probability of success should be zero. This will only work for assets in Task 1 because the program requires that all other task to be linear. Finally, make sure that the asset appears in the other tasks that it can support. This scheme will only work with option "e" since this is the only one in which the user can assign probabilities. Figure 2.6 and Table 2.6 is a simple example of how this will look.

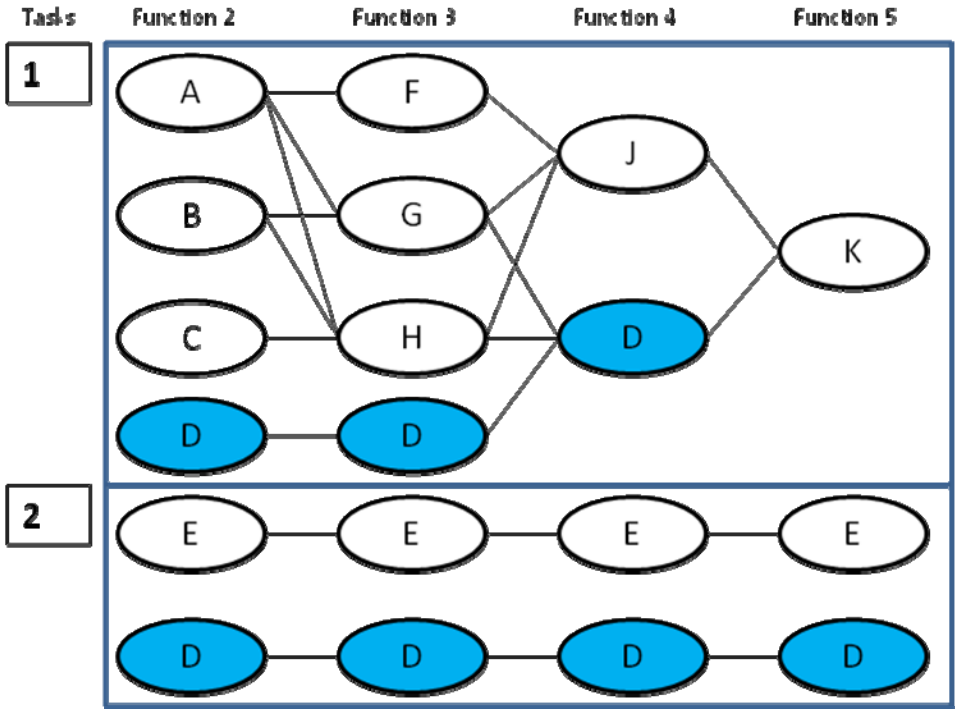


Figure 2.6: Example of Making a Decision in After Function 2

Table 2.6: Probabilities Used to Make a Decision After Function 2

Known Probabilities											
Function 2			Function 3			Function 4			Function 5		
Task 1											
$P(A_{1,2})$	0.99		$P(F_{1,3}\mid A_{1,2})$	0.59		$P(J_{1,4}\mid F_{1,3})$	0.87				
$P(B_{1,2})$	0.94		$P(G_{1,3}\mid A_{1,2})$	0.77		$P(J_{1,4}\mid G_{1,3})$	0.67				
$P(C_{1,2})$	0.66		$P(G_{1,3}\mid B_{1,2})$	0.56		$P(J_{1,4}\mid H_{1,3})$	0.53		$P(K_{1,5}\mid J_{1,4})$	0.97	
			$P(H_{1,3}\mid A_{1,2})$	0.98					$P(K_{1,5}\mid D_{1,4})$	0.61	
			$P(H_{1,3}\mid B_{1,2})$	0.80		$P(D_{1,4}\mid G_{1,3})$	0.89				
			$P(H_{1,3}\mid C_{1,2})$	0.78		$P(D_{1,4}\mid H_{1,3})$	0.64				
$\boldsymbol{P(D_{1,2})}$	**1.00**		$\boldsymbol{P(B_{1,3}\mid B_{1,2})}$	**1.00**		$\boldsymbol{P(D_{1,4}\mid D_{1,3})}$	**0.00**				
Task 2											
$P(E_{2,2})$	0.62		$P(E_{2,4}\mid E_{2,2})$	0.85		$P(E_{2,5}\mid E_{2,4})$	0.68		$P(E_{2,6}\mid E_{2,5})$	0.73	
$P(D_{2,2})$	0.72		$P(D_{2,4}\mid D_{2,2})$	0.63		$P(D_{2,5}\mid D_{2,4})$	0.74		$P(D_{2,6}\mid D_{2,5})$	0.91	

STRATCOM cannot disclose the exact details of the probabilities for obvious reasons. The probabilities used here solely for this demonstration. See Chapter 3 for more details.

3 Solving the SRATCOM Example

3.1 Unknown Probabilities

STRATCOM cannot disclose the actual probabilities of the assets involved or the exact network. To show how a user could use the program to study a network without the probabilities the following sections use the STATCOM example, Figure 1.1 through Figure 1.5, and the probabilities are sampled from a uniform distribution. This study could find the type of assets that the networks depend on and which assets have little to no effect on the final probability of success.

3.2 Ranges of Ten Percent

In the following sections, the probabilities are sampled from a uniform distribution with a ten percent range. It starts at only fifty percent and run as high as one hundred percent. First, all assets have probabilities sampled from the same range. After noticing almost no sensitivity to which assignments are made, following each section, the assignable assets were given a random probability from a uniform distribution with a range of ninety to one hundred for each range. Thirty runs of each range were preformed and the average probabilities are reported.

3.2.1 Probability Ranging from 50 to 60 Percent

Table A.1 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported is the final probability of the success of Task 1. Figure 3.1 is a graph of the data in Table A.1 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 4.57 percent.

3.2.2 Probability Ranging from 50 to 60 for Non-assignable and 90 to 100 Percent for Assignable Assets

Table A.2 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.2 is a graph of the data in Table A.2 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 5.22 percent, only 0.65 percent larger than when all the asset probabilities range from fifty to sixty percent. If assigning no assignable assets and only assigning Asset 11 to Task 1 is not considered than this range reduces to only 1.74 percent.

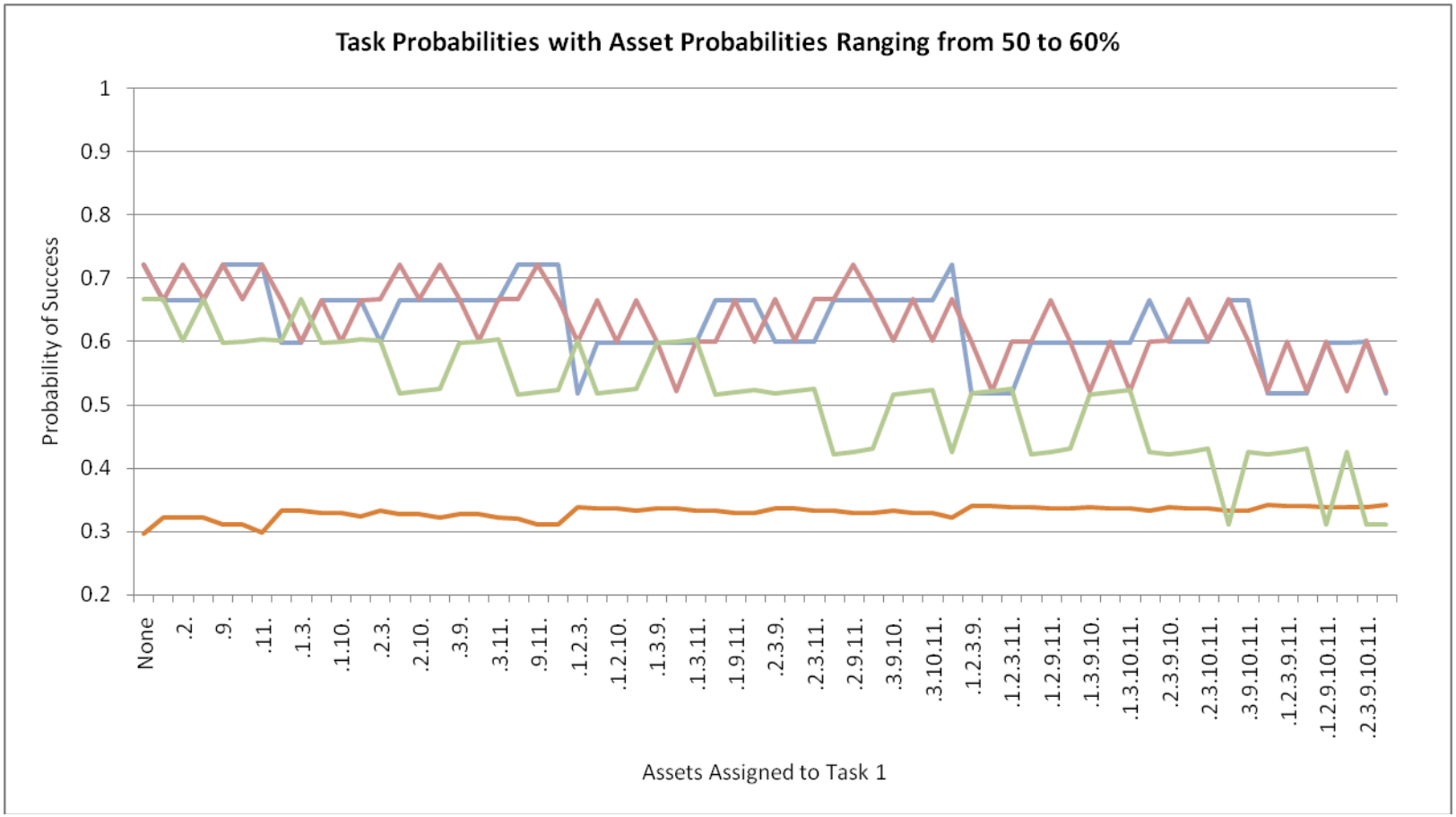


Figure 3.1: Graph of Task Probabilities with Assets Ranging from 50 to 60 Percent

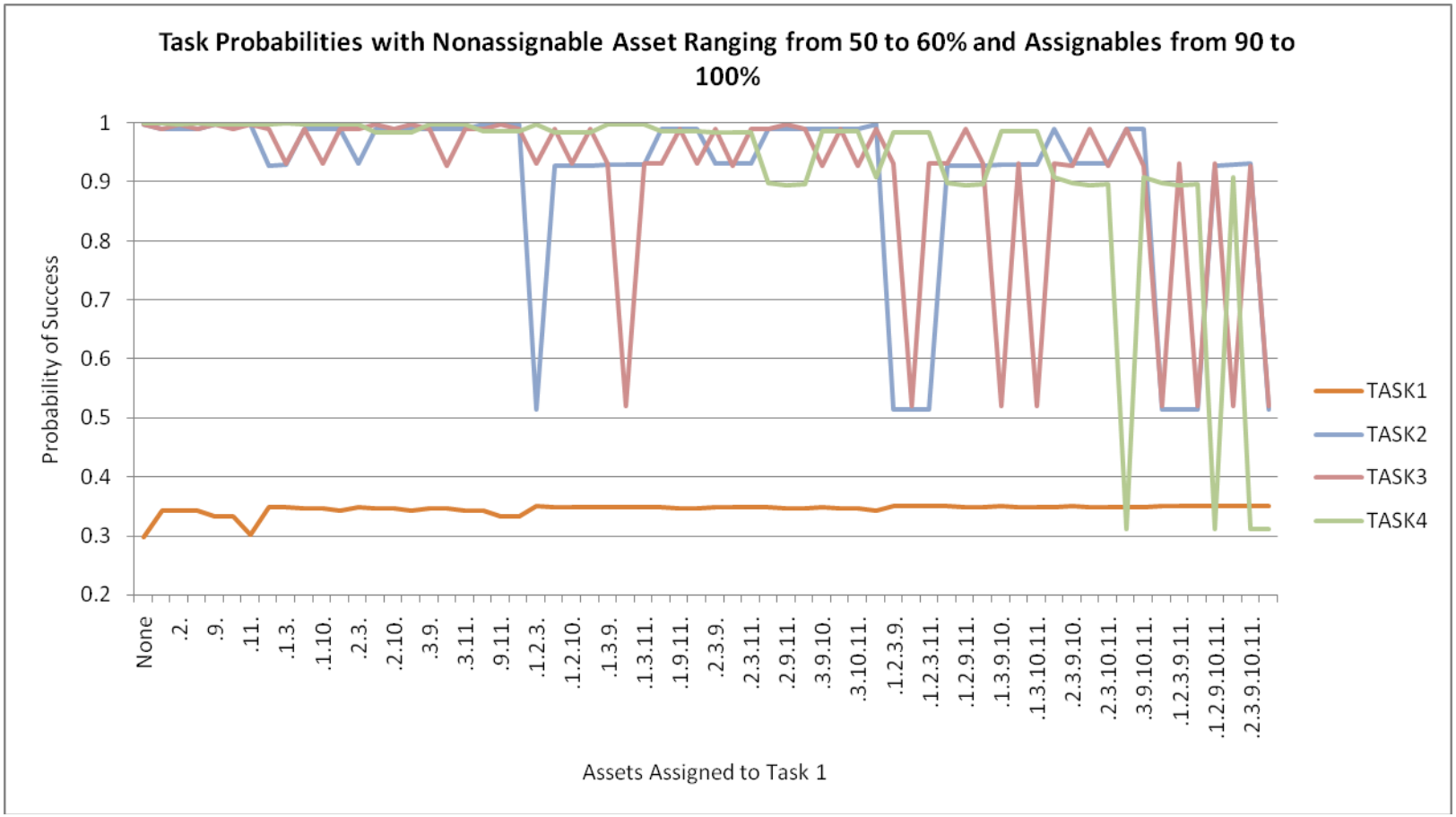


Figure 3.2: Graph of Task Probabilities with Non-assignable Assets Ranging from 50 to 60 and Others 90 to 100 Percent

3.2.3 Probability Ranging from 60 to 70 Percent

Table A.3 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.3 is a graph of the data in Table A.3 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 1.41 percent.

3.2.4 Probability Ranging from 60 to 70 for Non-assignable and 90 to 100 Percent for Assignable Assets

Table A.4 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.4 is a graph of the data in Table A.4 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is 1.43 percent, only 0.02 percent larger than when all the asset probabilities range from sixty to seventy percent. If assigning no assignable assets and only assigning Asset 11 to Task 1 is not considered than this range reduces to only 0.44 percent.

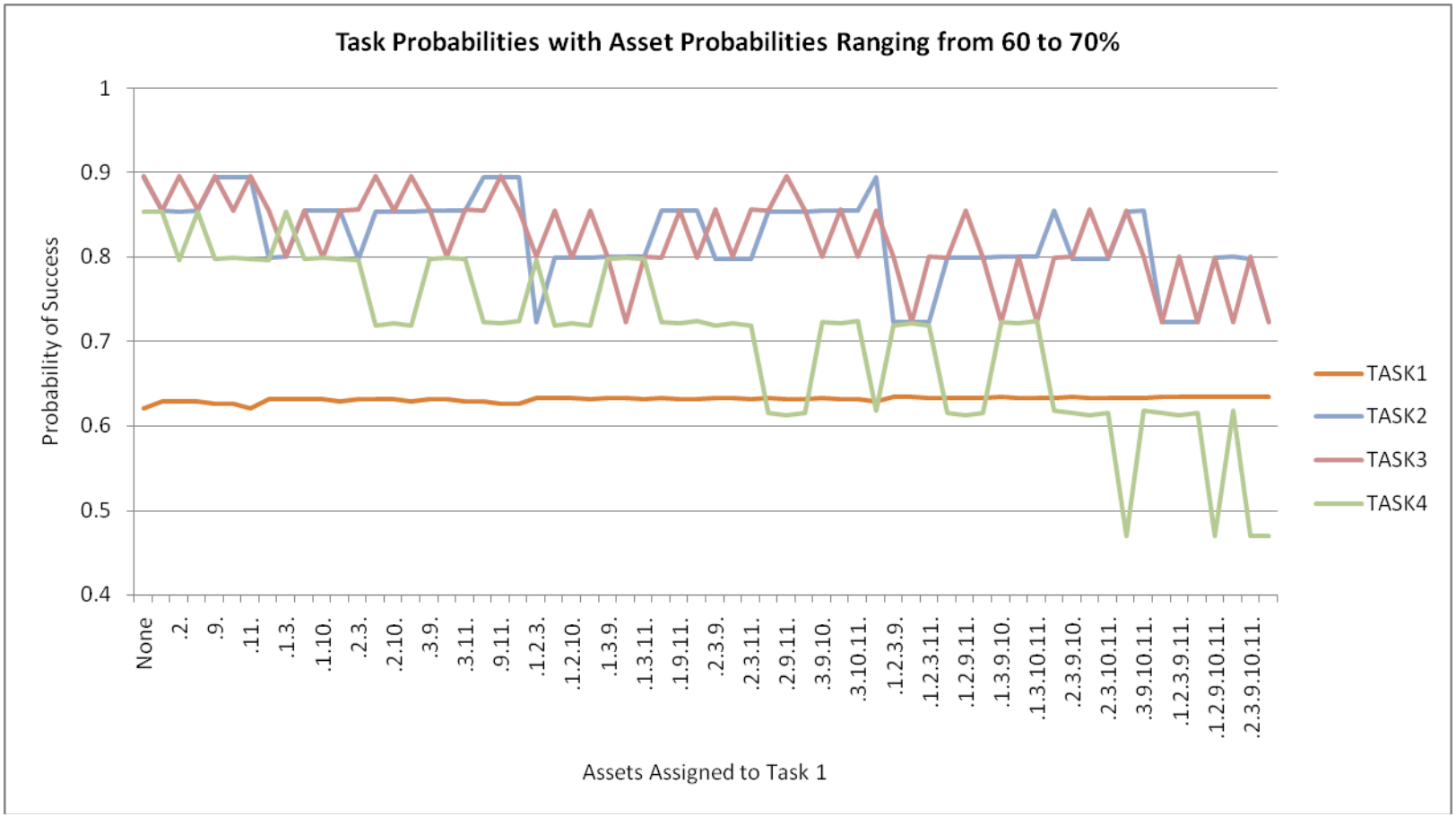


Figure 3.3: Graph of Task Probabilities with Assets Ranging from 60 to 70 Percent

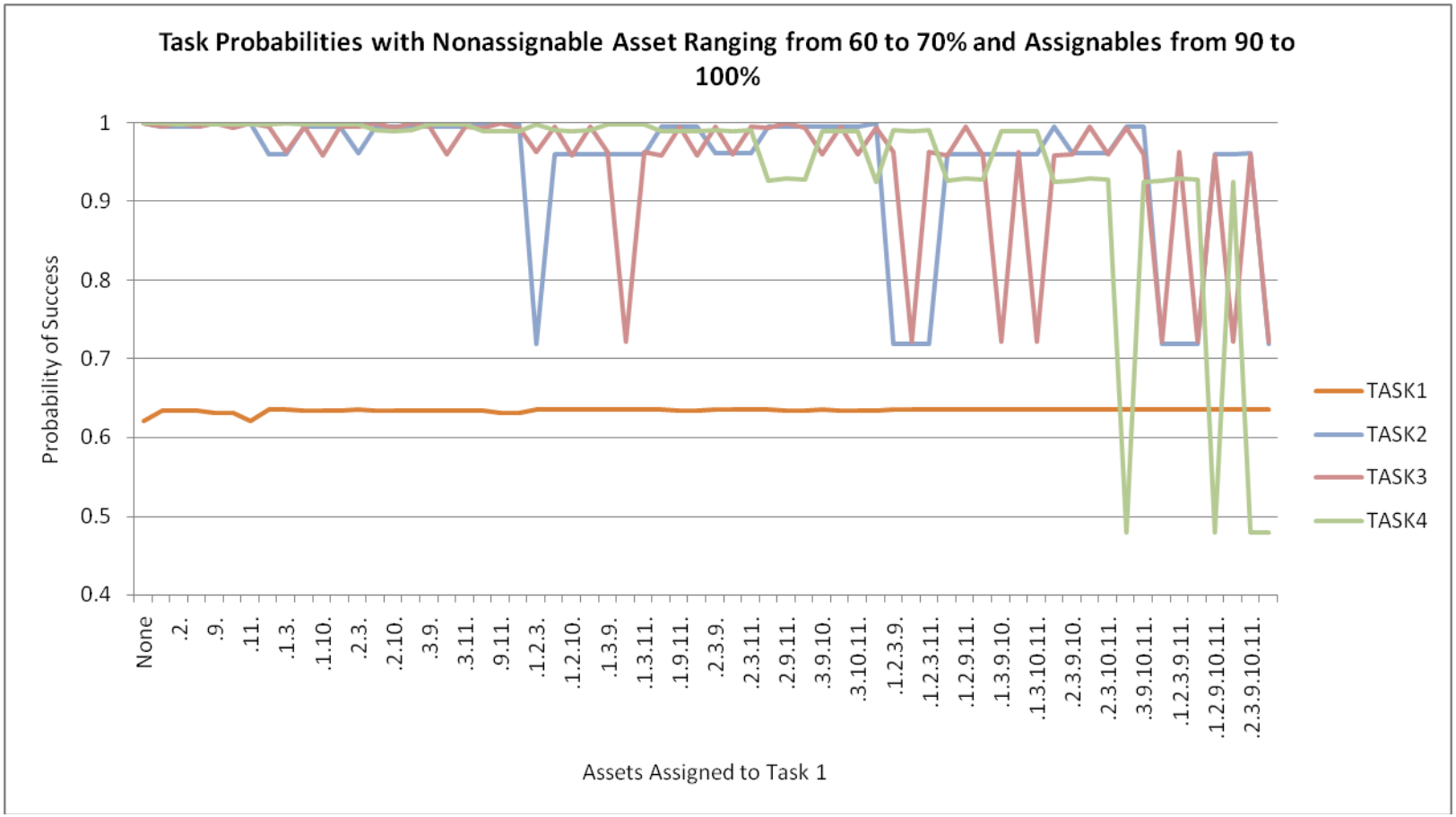


Figure 3.4: Graph of Task Probabilities with Non-assignable Assets Ranging from 60 to 70 and Others 90 to 100 Percent

3.2.5 Probability Ranging from 70 to 80 Percent

Table A.5 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.5 is a graph of the data in Table A.5 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 0.10 percent.

3.2.6 Probability Ranging from 70 to 80 for Non-assignable and 90 to 100 Percent for Assignable Assets

Table A.6 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.4 is a graph of the data in Table A.6 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is 0.12 percent, only 0.02 percent larger than when all the asset probabilities range from sixty to seventy percent. If assigning no assignable assets and only assigning Asset 11 to Task 1 is not considered than this range reduces to only 0.04 percent

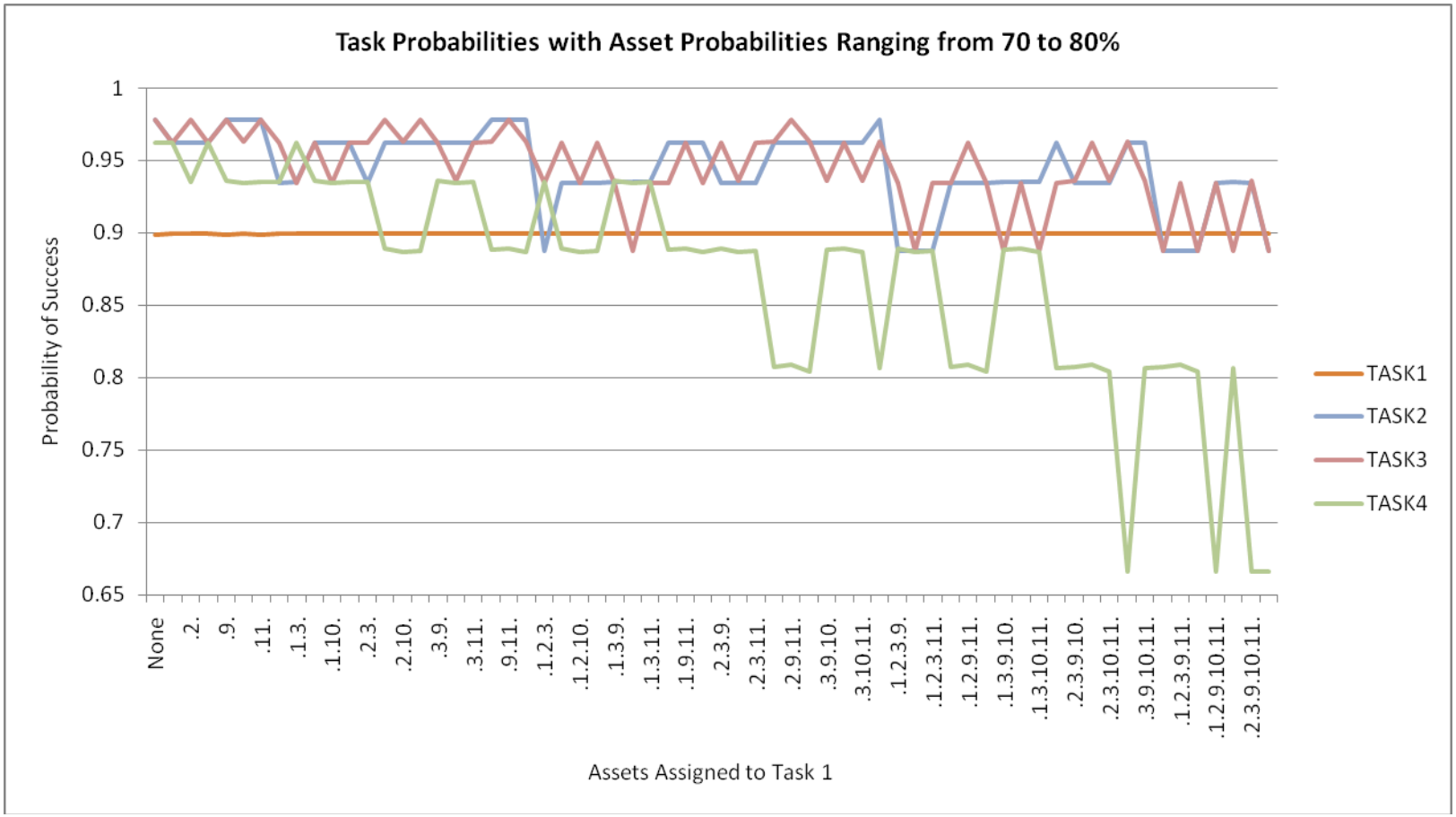


Figure 3.5: Graph of Task Probabilities with Assets Ranging from 70 to 80 Percent

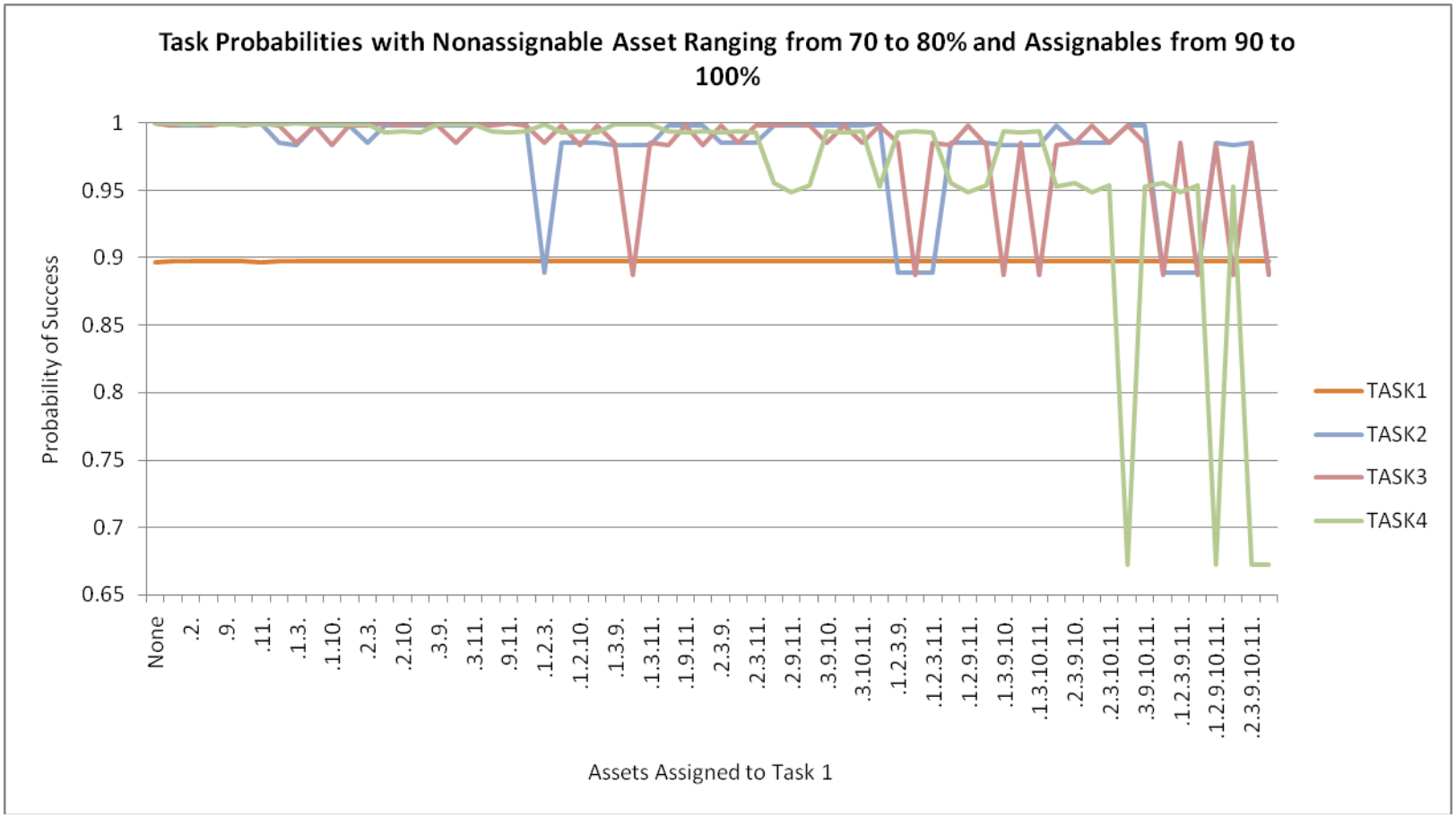


Figure 3.6: Graph of Task Probabilities with Non-assignable Assets Ranging from 70 to 80 and Others 90 to 100 Percent

3.2.7 Probability Ranging from 80 to 90 Percent

Table A.7 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.7: Graph of Task Probabilities with Assets Ranging from 80 to 90 Percent is a graph of the data in Table A.7 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 0.0009 percent.

3.2.8 Probability Ranging from 80 to 90 for Non-assignable and 90 to 100 Percent for Assignable Assets

Table A.8 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.8 is a graph of the data in Table A.8 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is 0.0010 percent, only 0.0001 percent larger than when all the asset probabilities range from sixty to seventy percent. If assigning no assignable assets and only assigning Asset 11 to Task 1 is not considered than this range reduces to only 0.0003 percent.

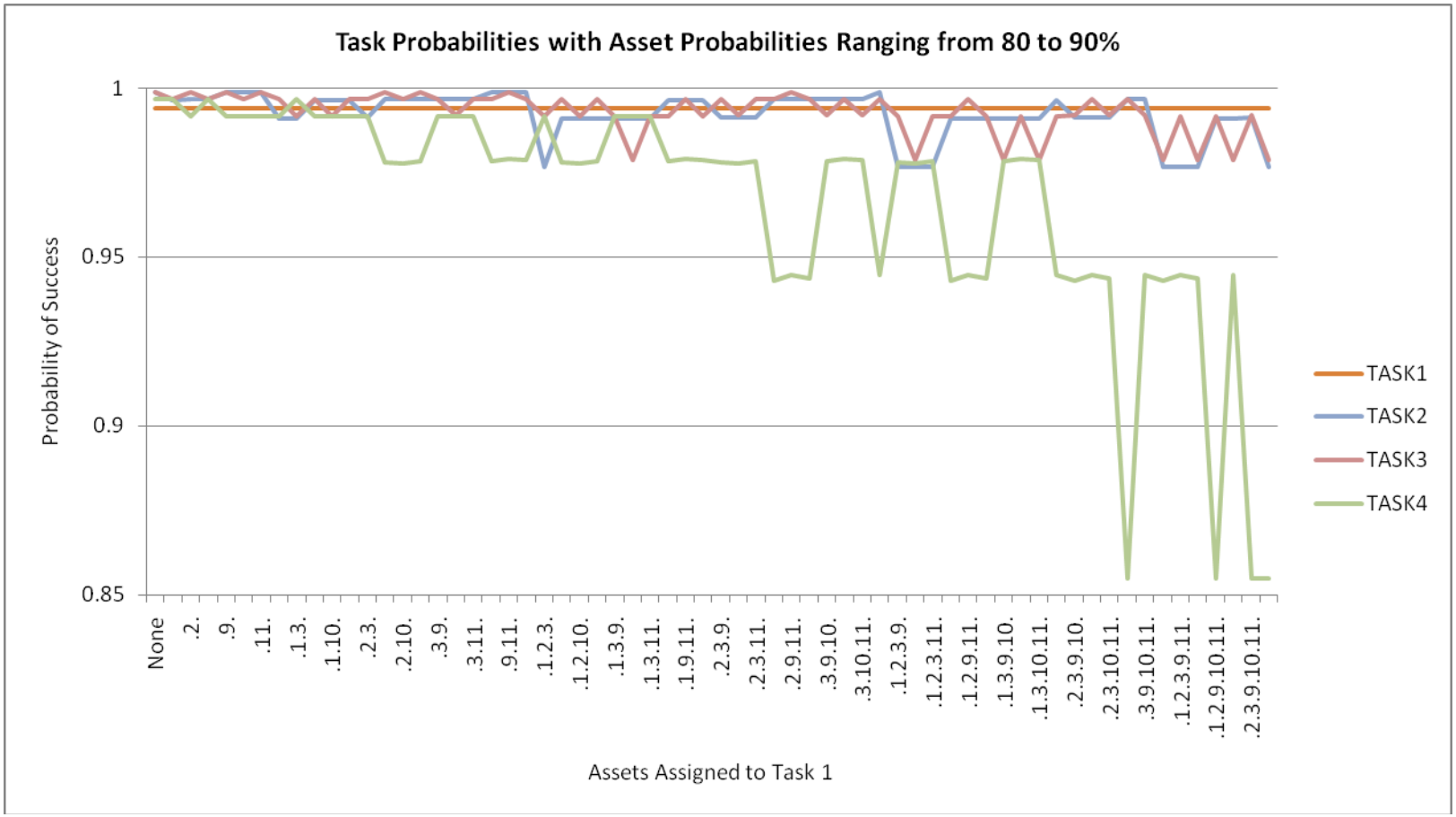


Figure 3.7: Graph of Task Probabilities with Assets Ranging from 80 to 90 Percent

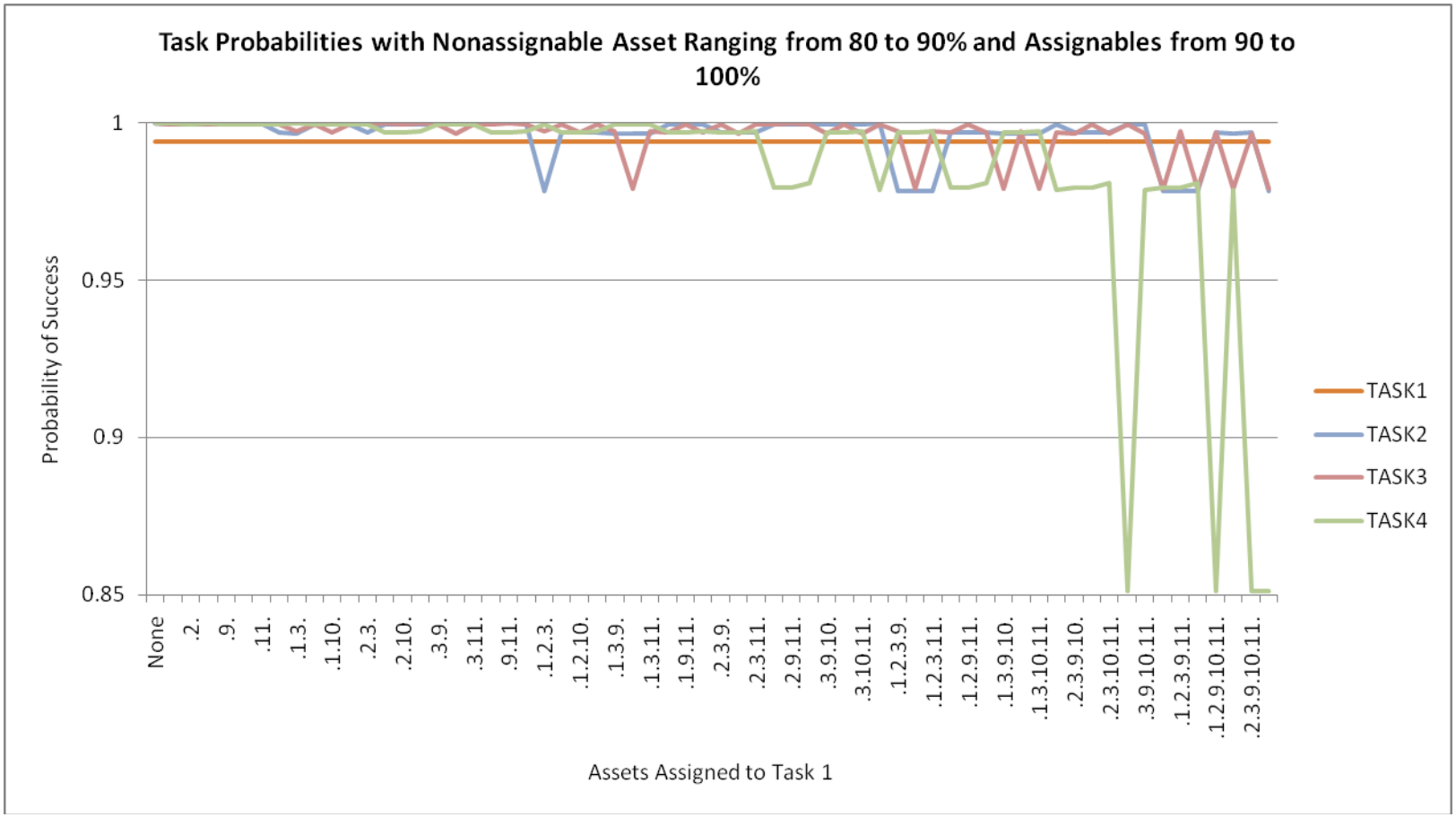


Figure 3.8: Graph of Task Probabilities with Non-assignable Assets Ranging from 80 to 90 and Others 90 to 100 Percent

3.2.9 Probability Ranging from 90 to 100 Percent

Table A.9 shows the results when the assets' probability of success is randomly generated from a uniform density function and thirty runs are averaged. The program gives the results of each asset in the final function of Task 1; reported here is the final probability of the success of Task 1. Figure 3.9 is a graph of the data in Table A.9 to make comparing the different combinations of the assignable easier. Notice that the difference between the highest, all assignable assets, and the lowest probability of success of Task 1, no assignable assets, is only 0.00000008 percent.

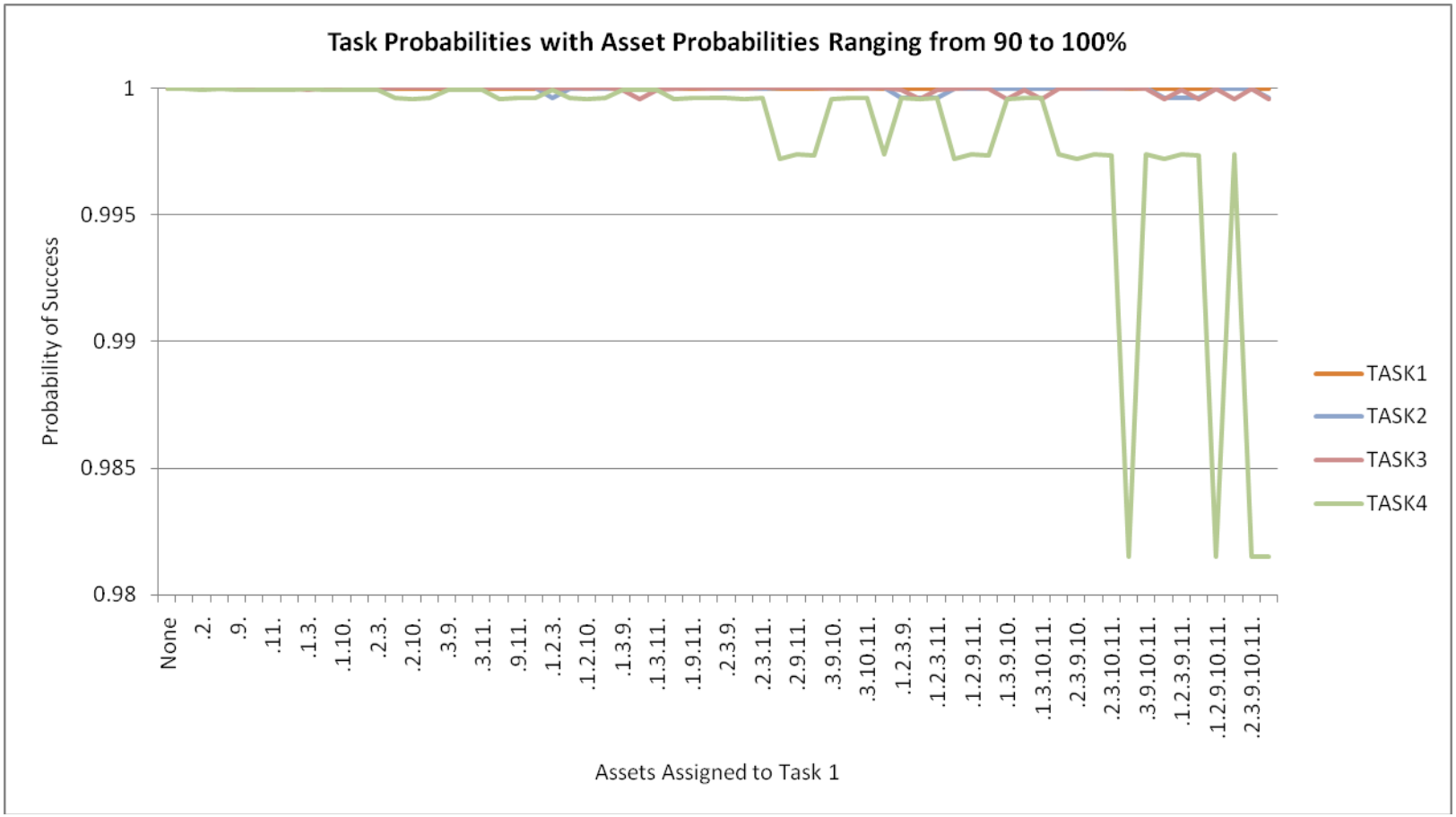


Figure 3.9: Graph of Task Probabilities with Assets Ranging from 90 to 100 Percent

3.2.10 Summary of Ten Percent Ranges

The shape of each of the graphs is the same. As the probability of success is increased the probabilities of success of the different combinations becomes less variable and closers to one. Graphed in Figure 3.10 is the difference between the highest and lowest probability of success is in Table 3.1. The standard deviations were examined and the graph of them is identical to the graph in Figure 3.10, but on a smaller scale. Depending on the actual probabilities of success of the real assets, STRATCOM may need to be determined at what point it is worth placing an additional asset into a network. If the probabilities are in the 80 – 90 range then the difference between have all the assets assigned to Task 1 and none of them assigned to Task 1 is only 0.00088 percent. The other tasks have differences of 2.189, 2.013, and 14.199 percent, respectively. This inability to affect the final probability is an indicator of the robustness of the network and what causes a network to become this way needs research. The simple example in Figure 2.1 even starts to show signs of this in Task 1 when the probabilities range from 53 to 99 percent.

Table 3.1: Differences between the High and Low Probabilities within each Range

The Difference Between the Highest and Lowest Probability within each Range				
Range (%)	TASK1	TASK2	TASK3	TASK4
50-60	0.0457080548	0.2036756477	0.1994500306	0.3552500391
60-70	0.0140558127	0.1721214281	0.1720157880	0.3835738111
70-80	0.0010354543	0.0906239225	0.0910142344	0.2969570321
80-90	0.0000088306	0.0218933714	0.0201292061	0.1419861561
90-100	0.0000000008	0.0003803049	0.0004358103	0.0184970435

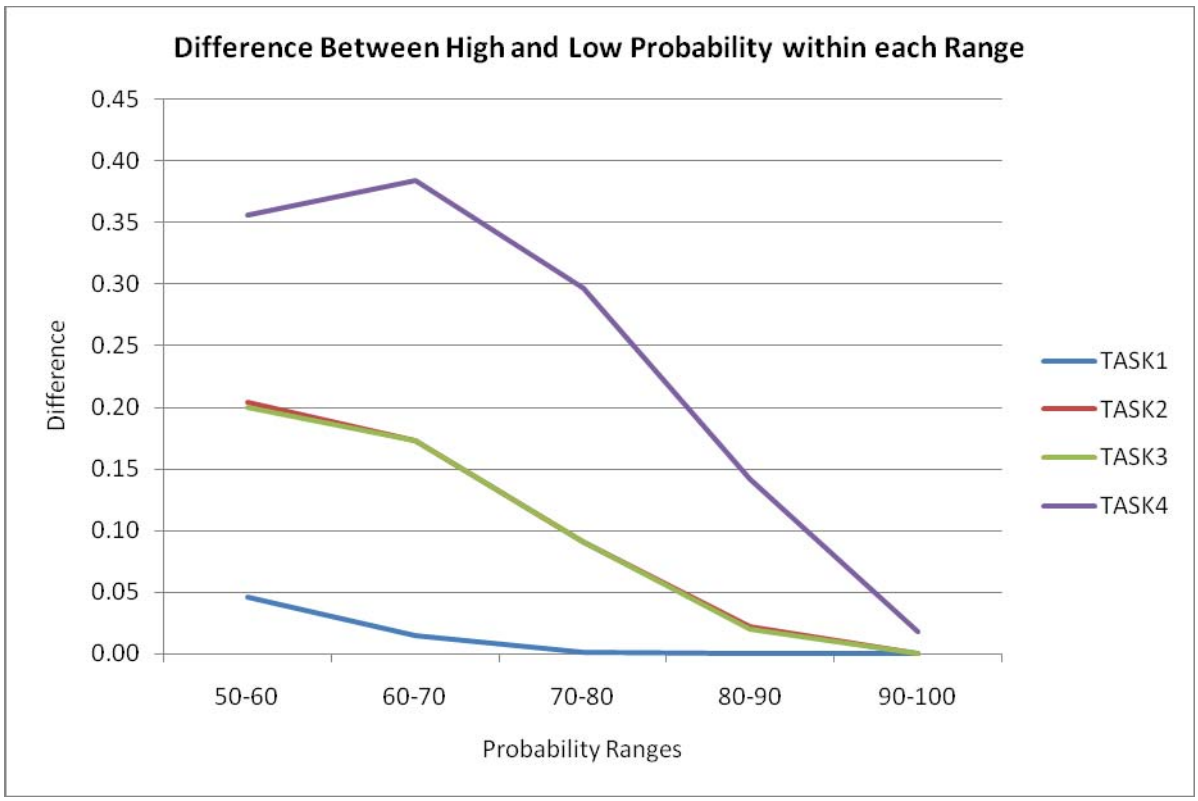


Figure 3.10: Graph of the Differences Between the High and Low Probability within each Range

3.2.11 Summary of Improving the Assignable Assets

When the assignable assets have a higher probability of success, the sensitivity of the network is higher. Table 3.2 shows the range of the probabilities for each range of probabilities for Task 1 and Figure 3.11 graphs this difference. However, this change did not have a major effect on the sensitivity of the network.

Table 3.2: Task 1, Comparing STRATCOM's Network Sensitivity with Assets Having All the Same to Assignables with Better Probabilities

Task 1			
Range (%)	All Same	Better Assignables	Difference
50-60	0.0457080548	0.052166519	0.0064584642
60-70	0.0140558127	0.0143376232	0.0002818105
70-80	0.0010354543	0.0011944351	0.0001589808
80-90	0.0000088306	0.0000101468	0.0000013162
90-100	0.0000000008	0.0000000008	0.0000000000

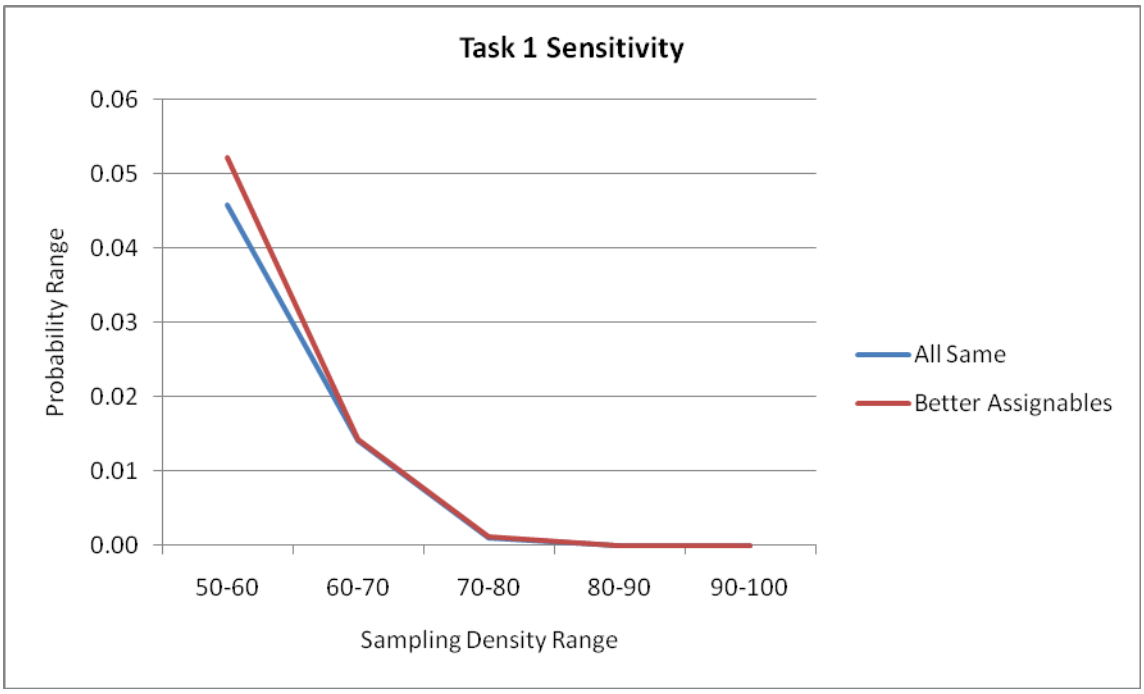


Figure 3.11: Task 1 Sensitivity

3.3 Examining a 20 Percent Range

Ranges of 20 percent were also used just as the 10 percent ranges and there were no real changes. Figure 3.12 is a graph of Task 1 for all the different ranges. Notice that the 20 percent ranges fall between the 10 percent ranges just as one would expect. The differences between the maximum and minimum probabilities of success were also calculated and nothing unexpected was found.

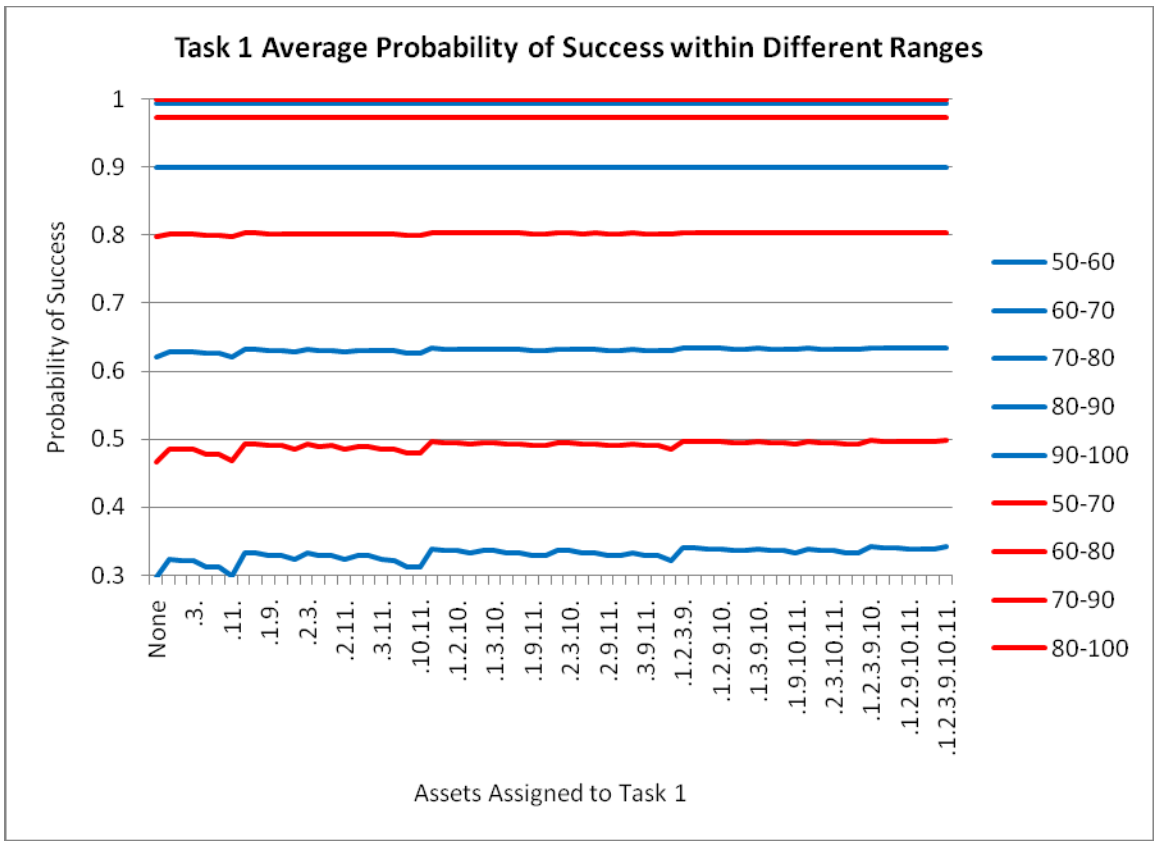


Figure 3.12: All of the Task 1 Probabilities for all Ranges

3.4 Enemy Degradation of a Network

Due to the insensitivity of the STRATCOM example when reassigning assets, a study of what happens when an enemy destroys an asset was conducted. To conduct this study the STRATCOM example was given random probabilities, ranging from 70-80% for the nonassignable assets and 90-100% for the assignable assets. These probabilities are detailed in Appendix C. For this study assume the enemy could destroy any of the assets that could be assigned to perform Function 2 of Task 1. However, the enemy does not know the assignments of the asset to the tasks. The next sections look at Task 1 total probability of success with an enemy destroying all combinations of these assets starting with no assignable assets assigned to Task 1 and ending with all assignable assets assigned to Task 1. Contained in Appendix D is a detailed description of how this study was conducted so that it can be repeated by STRATCOM with real data.

3.4.1 No Assignable Assets Assigned to Task 1

Since assigning the assignable assets to Task 1 does not greatly affect Task 1 probability of success, assign all assignable assets to the other tasks. The network looks like Figure 3.14, below. The probability of success for Task 1 is 91.72%. Table 3.3, below lists the minimum, average, and maximum probability of success when the enemy destroys a number of the assets. Figure 3.13 graphs this information. The maximum is where the enemy destroys the assets not assigned to Task 1 first and then destroys the assets that contribute the least to the total probability of success of Task 1. The average is an average of all possible ways to destroy that particular number of assets. The minimum is if the enemy destroys the assets that contribute the most to the probability of success first.

Table 3.3: Probability of Success of Task 1 with No Assignable Assets after an Enemy Destroys Assets

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9172	0.8296	0.7374	0.3929	0.0734	0.0000
Average	0.9172	0.8951	0.8723	0.8467	0.8148	0.7724
Maximum	0.9172	0.9172	0.9172	0.9172	0.9172	0.9172
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.7143	0.6353	0.5296	0.3919	0.2168	0.0000
Maximum	0.9172	0.9169	0.9008	0.8353	0.7547	0.0000

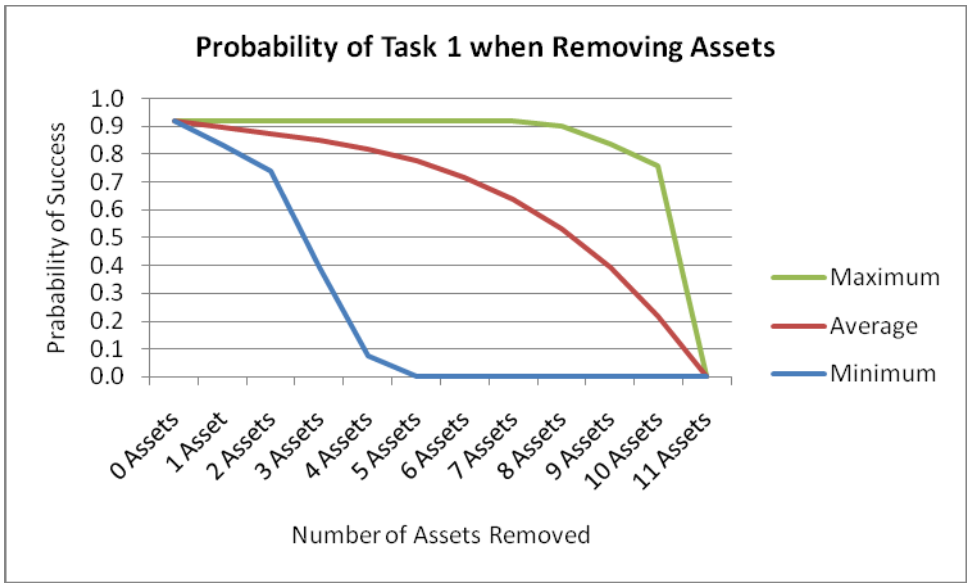


Figure 3.13: Graph of the Probability of Success of Task 1 with No Assignable Assets after an Enemy Destroys Assets

Note that if an enemy knows which assets will do the most damage after destroying two assets the probability of success drops off dramatically. Table 3.4 list the assets, when destroyed do the most damage to the probability of success of Task 1.

Table 3.4: The Assets Most Critical to the Probability of Success of Task 1 with no Assignable Assets

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	4, 5, 6
4 Assets	4, 5, 6, 8
5 Assets	4, 5, 6, 8, 7

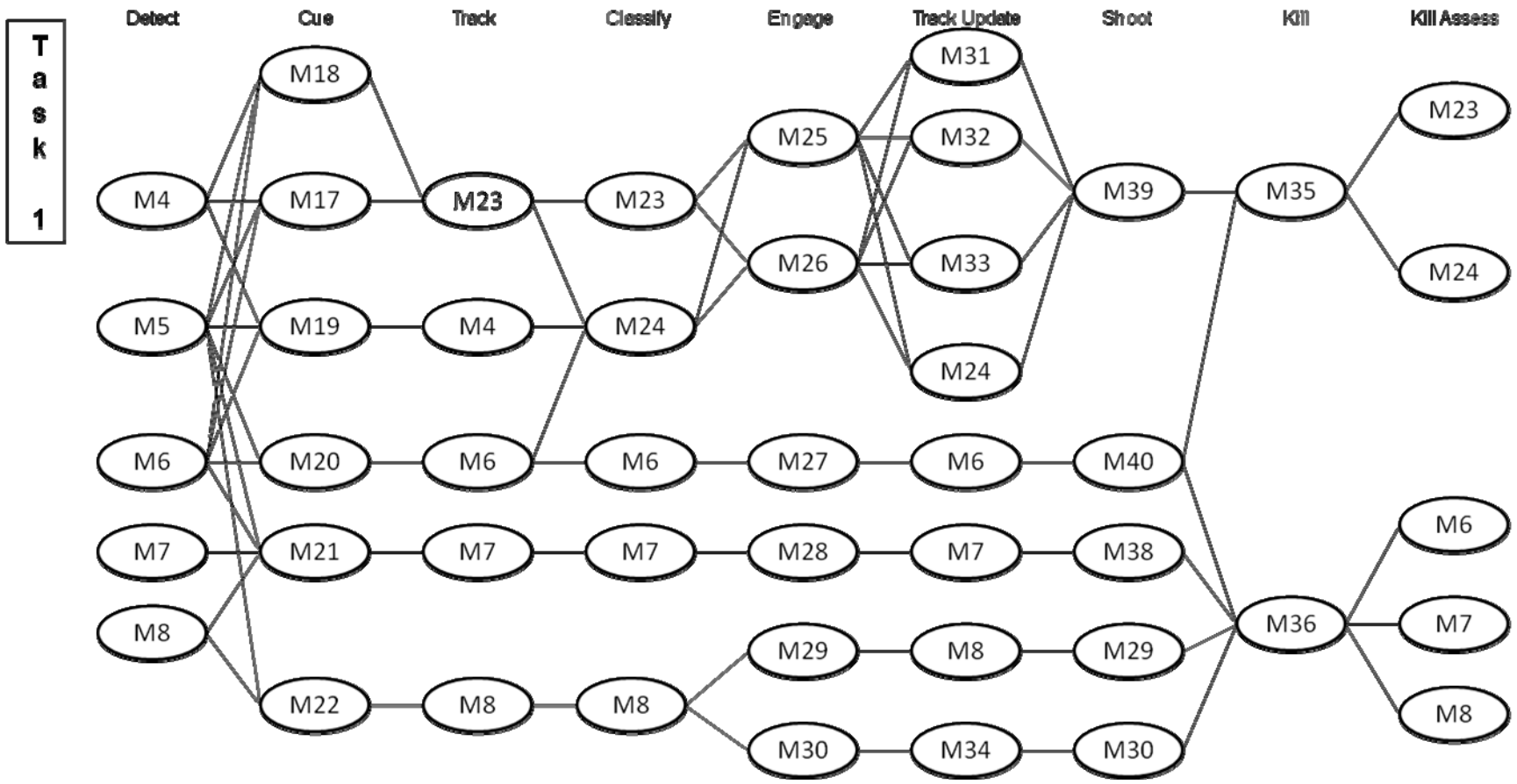


Figure 3.14: Task 1 with No Assignable Assets

3.4.2 One Assignable Assets Assigned to Task 1

Now assign the asset that changes the average of Tasks 2, 3, and 4 to Task 1, Asset 11. The network looks like Figure 3.16, below. The probability of success for Task 1 is 91.72%. Table 3.5, below lists the minimum, average, and maximum probability of success when the enemy destroys a number of the assets. Figure 3.15 graphs this information. The maximum is where the enemy destroys the assets not assigned to Task 1 first and then destroys the assets that contribute the least to the total probability of success of Task 1. The average is an average of all possible ways to destroy that particular number of assets. The minimum is if the enemy destroys the assets that contribute the most to the probability of success first.

Table 3.5: Probability of Success of Task 1 with Asset 11 Assigned after an Enemy Destroys Assets

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9172	0.8297	0.7376	0.7093	0.3929	0.0734
Average	0.9172	0.8951	0.8724	0.8495	0.8255	0.7972
Maximum	0.9172	0.9172	0.9172	0.9172	0.9172	0.9172
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.7589	0.7016	0.6133	0.4789	0.2809	0.0000
Maximum	0.9172	0.9169	0.9008	0.8353	0.7547	0.0000

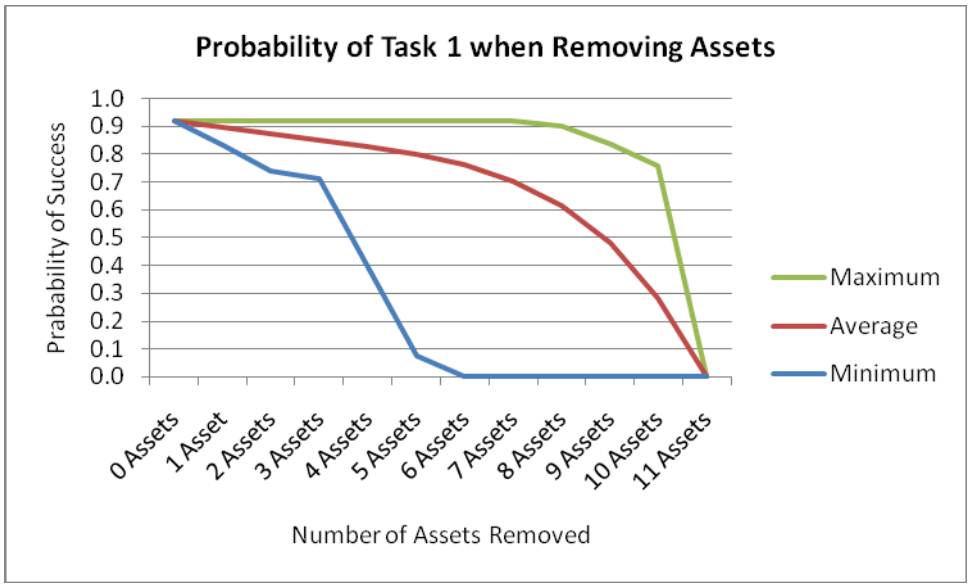


Figure 3.15: Graph of the Probability of Success of Task 1 with Asset 11 Assigned after an Enemy Destroys Assets

Note that if an enemy knows which assets will do the most damage after destroying three assets the probability of success drops off dramatically. This is one asset more then with no assets assigned to Task 1 as in section 3.4.1. Table 3.6 list the assets, when destroyed do the most damage to the probability of success of Task 1.

Table 3.6: The Assets Most Critical to the Probability of Success of Task 1 with Asset 11 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	4, 5, 6, 11
5 Assets	4, 5, 6, 11, 8
6 Assets	4, 5, 6, 11, 8, 7

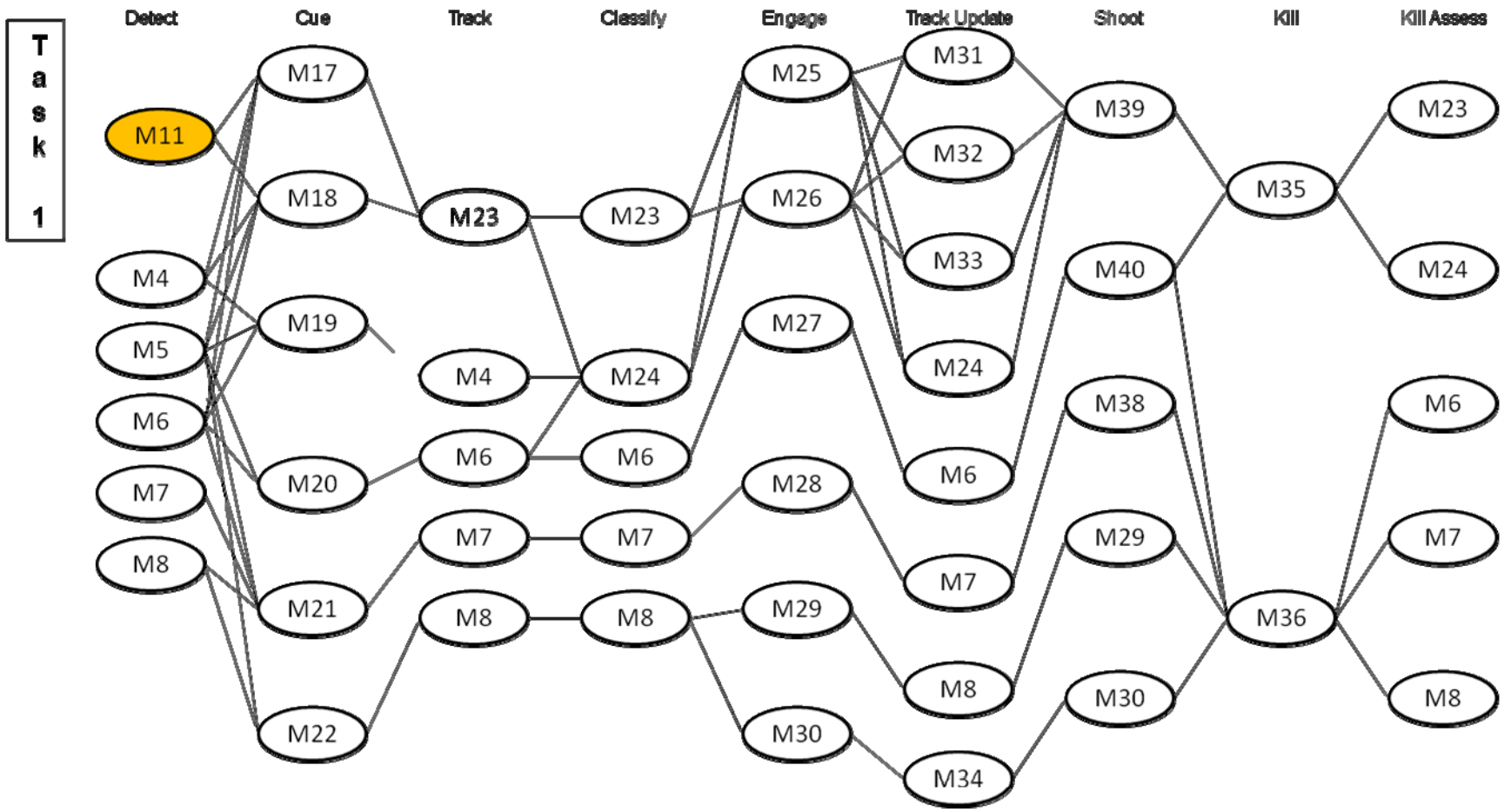


Figure 3.16: Task 1 with Asset 11 Assigned

3.4.3 Two through All Six Assignable Assets Assigned to Task 1

This pattern of shifting the dramatic drop in Task 1 probability continues to shift one asset for each asset assigned to Task 1. Figure 3.17 through Figure 3.21 depict this. The data used to generate these figures is located in Appendix C. With this analysis of a network, a decision maker can see and make decision about the number of assets and which assets are most critical to that network. This analysis is only valid for the STRATCOM example with the probabilities assigned as shown in Appendix C. The assets that are most critical in this study may change if the probabilities of success are different.

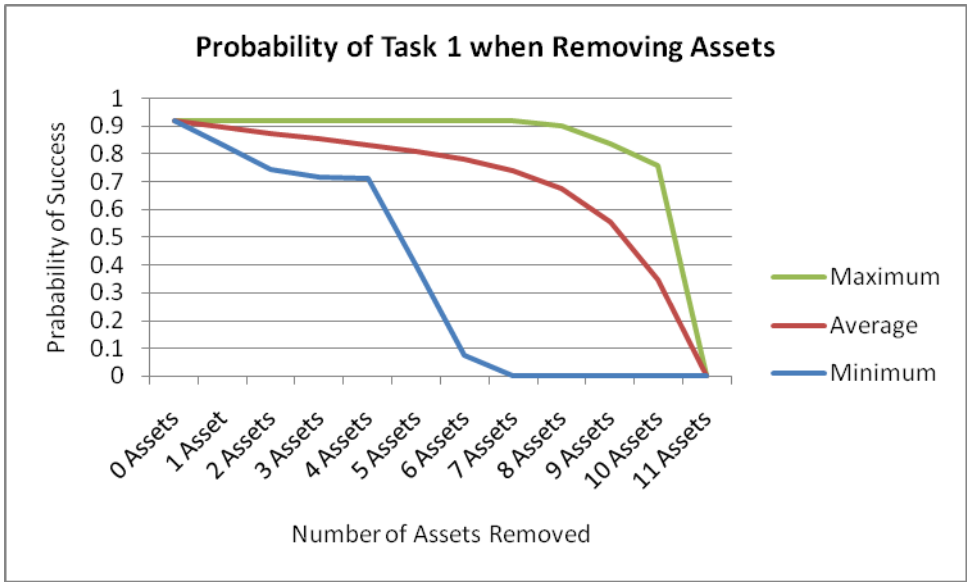


Figure 3.17: Graph of the Probability of Success of Task 1 with Assets 11 and 2 Assigned after an Enemy Destroys Assets

Table 3.7: The Assets Most Critical to the Probability of Success of Task 1 with Assets 11 and 2 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	8, 6, 7, 2
5 Assets	6, 2, 4, 5, 11
6 Assets	6, 2, 4, 5, 11, 8
7 Assets	6, 2, 4, 5, 11, 8, 7

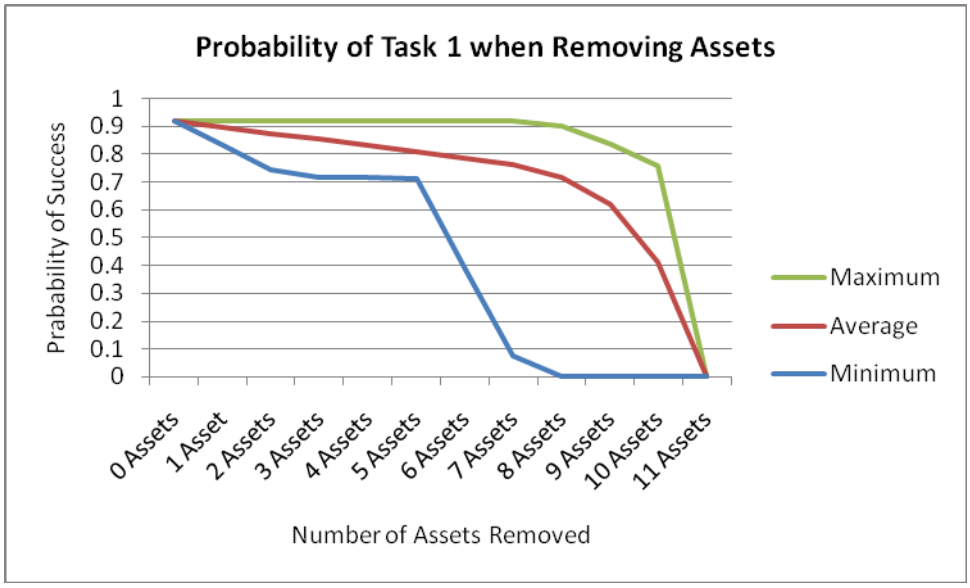


Figure 3.18: Graph of the Probability of Success of Task 1 with Assets 11, 2, and 10 Assigned after an Enemy Destroys Assets

Table 3.8: The Assets Most Critical to the Probability of Success of Task 1 with Assets 11, 2, and 10 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	8, 6, 7, 2
5 Assets	8, 6, 7, 2, 10
6 Assets	6, 2, 10, 4, 5, 11
7 Assets	6, 2, 10, 4, 5, 11, 8
8 Assets	6, 2, 10, 4, 5, 11, 8, 7

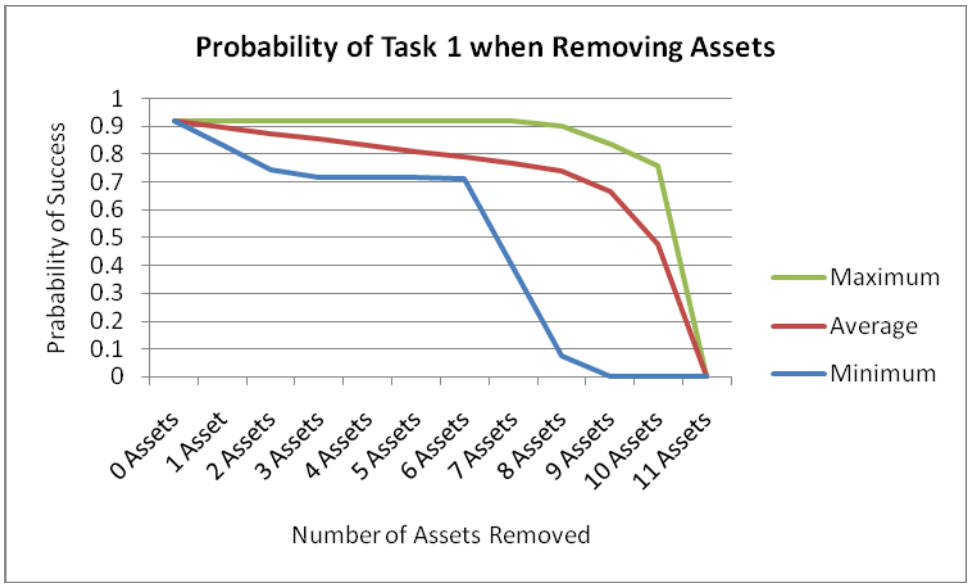


Figure 3.19: Graph of the Probability of Success of Task 1 with Asset 11, 2, 10, and 3 Assigned after an Enemy Destroys Assets

Table 3.9: The Assets Most Critical to the Probability of Success of Task 1 with Assets 11, 2, 10, and 3 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	8, 6, 7, 3
5 Assets	8, 6, 7, 3, 2
6 Assets	8, 6, 7, 3, 2, 10
7 Assets	6, 3, 2, 10, 4, 5, 11
8 Assets	6, 3, 2, 10, 4, 5, 11, 8
9 Assets	6, 3, 2, 10, 4, 5, 11, 8, 7

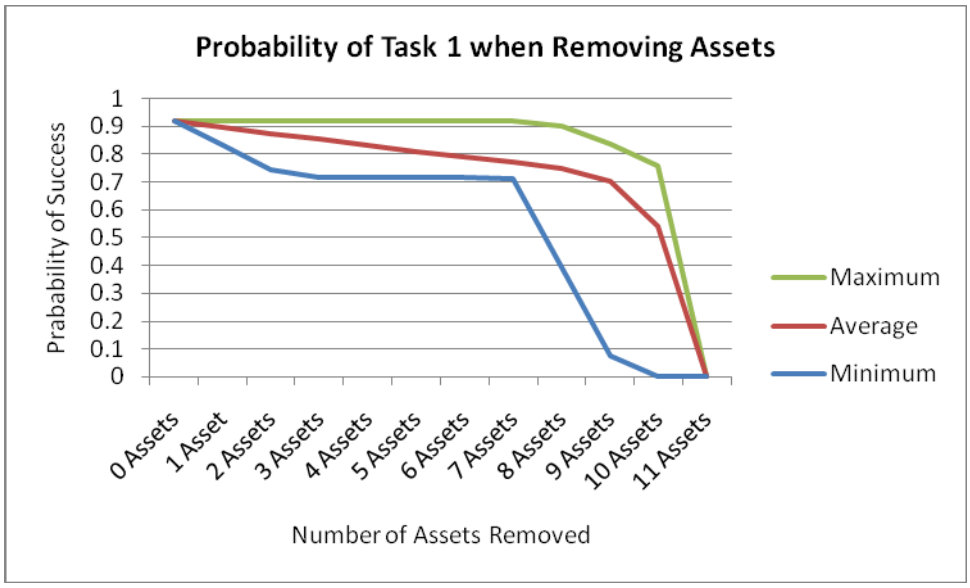


Figure 3.20: Graph of the Probability of Success of Task 1 with Asset 11, 2, 10, 3, and 1 Assigned after an Enemy Destroys Assets

Table 3.10: The Assets Most Critical to the Probability of Success of Task 1 with Assets 11, 2, 10, 3, and 1 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	8, 6, 7, 1
5 Assets	8, 6, 7, 1, 3
6 Assets	8, 6, 7, 1, 3, 2
7 Assets	8, 6, 7, 1, 3, 2, 10
8 Assets	6, 1, 3, 2, 10, 4, 5, 11
9 Assets	6, 1, 3, 2, 10, 4, 5, 11, 8
10 Assets	6, 1, 3, 2, 10, 4, 5, 11, 8, 7

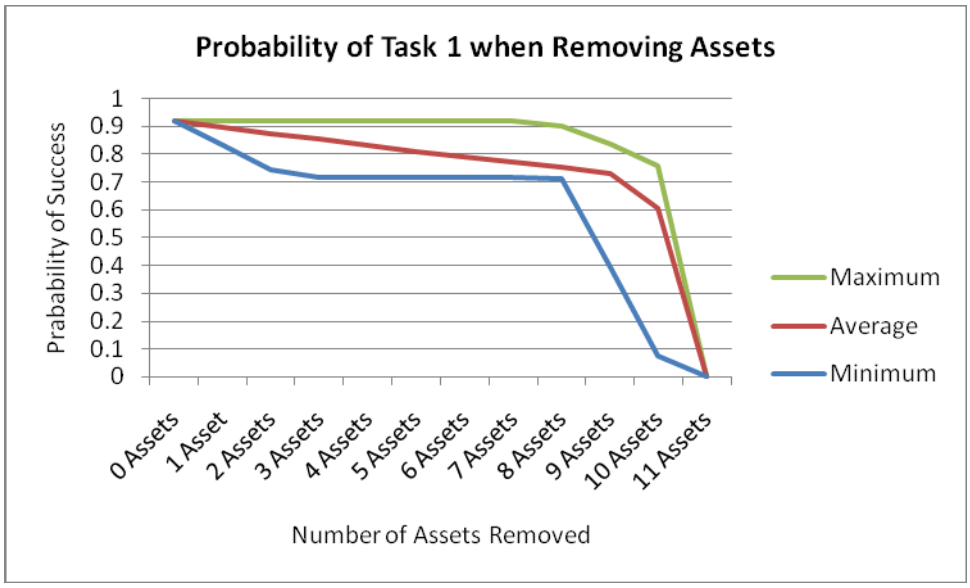


Figure 3.21: Graph of the Probability of Success of Task 1 with Asset 11, 2, 10, 3, 1, and 9 Assigned after an Enemy Destroys Assets

Table 3.11: The Assets Most Critical to the Probability of Success of Task 1 with Assets 11, 2, 10, 3, 1, and 9 Assigned

# of Assets	Asset #
1 Asset	8
2 Assets	8, 6
3 Assets	8, 6, 7
4 Assets	8, 6, 7, 1
5 Assets	8, 6, 7, 1, 3
6 Assets	8, 6, 7, 1, 3, 2
7 Assets	8, 6, 7, 1, 3, 2, 9
8 Assets	8, 6, 7, 1, 3, 2, 9, 10
9 Assets	6, 1, 3, 2, 9, 10, 4, 5, 11
10 Assets	6, 1, 3, 2, 9, 10, 4, 5, 11, 8
11 Assets	6, 1, 3, 2, 9, 10, 4, 5, 11, 8, 7

3.5 The STRATCOM Example Conclusions

The network seems to be insensitive to the change in assignable assets. This is especially true when the probability of success of the assets is high. To show some sensitivity the elimination of assets was exploded and this analysis may give a decision maker something to consider when allocating assets. However, only one particular network with particular

probabilities was studied. Furthermore, the entire thesis makes a major assumption that all of the assets have, nearly, the same probability of success. Finally, this study does not consider the possibility of multiple events within a task and whether or not an asset can handle more than one event simultaneously.

4 Future Work

4.1 Regression Analysis

A regression analysis of the effects of adding different types of assets to a network may lead to some useful findings. The factors may include the functions that the asset is in, number of connections to the previous and next function, and the probability of success of the asset. A random sample of possible networks is needed to conduct this study. The current program can only generate random probabilities for a given network. This would need to be added or facilitated from another source. A study of this type may allow the classification of a network as sensitive and insensitive to change. This then would allow STRATCOM to focus their efforts on the sensitive networks. This may also reveal the types of asset that are best candidates for reassignment to another lower probability of success network.

4.2 Effect of Asset Probabilities within the Degradation Study

The degradation study in this thesis only considers one network with particular probabilities. A study on the sensitivity of the asset probabilities this would reveal at what probabilities which assets are essential to the network.

4.3 Increasing the Speed of the Program

The speed of the program was never a concern in this study. This study focused on how to solve the problem to the satisfaction of STRATCOM. The next step would be to improve the speed of the program and to see if total enumeration can be completed within the time constraints that STRATCOM has.

4.4 STRATCOM Improvements and Uses for this Program

The following studies require information that STRATCOM seemed unwilling to disclose in the public forum of the study for understandable reasons.

4.4.1 Changing the Density Function Used to Generate Probabilities

One aspect of the problem that STRATCOM left unclear was if the assets of a particular function would have a relatively different range of probabilities then the others. An attempt to study the effects of this was attempted by allowing a different range of random probabilities for the assets in Function 2 of Task 1 than the others. This change did not affect the sensitivity of the probability of Task 1 success, so nothing further was studied. However, if STRATCOM is using this program to study a particular network that they wanted to randomly generate probabilities. They may need to single out particular functions in the same manner that the program singles out Task 1 Function 2. This would allow them to run random probabilities that more closely reflect reality.

4.4.2 Current Network Analysis

STRATCOM can use the current program to analyze its current networks. If a network is too robust then assets could be reassigned to less robust networks. This kind of study could reveal both strengths and weaknesses. Knowing this, decisions can be made to improve networks and the security of the American public. This information should not be limited to moving and trading assets, but it could help with acquisition of new assets. A decision maker would know the effect of have an asset with certain connections in a certain function with a certain probability of success and the effects of not meeting or exceeding the desired requirements.

REFERENCES

1. Arney, K. M. (2008). Global Sensor Management: Allocation of Military Surveillance Assets (Master's thesis, North Carolina State University, 2008).

2. Dulin, J.L. (2009). Global Sensor Management: Real-Time Reallocation of Military Assets among Competing Tasks and Functions (Doctoral dissertation, North Carolina State University, 2009).

3. Ross, S. M. (2007). *Introduction to Probability Models* (9th Ed.). Burlington, MA: Academic Press.

4. Scheaffer, R. L. (1995). *Introduction to Probability and Its Applications* (2nd Ed.). Belmont, CA: Duxbury Press.

5. Wackerly, D. D., Mendenhall III, W., & Scheaffer, R. L. (2002). *Mathematical Statistics with Applications* (6th Ed.). Pacific Grove, CA: Duxbury.

6. Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2002). *Probability & Statistics for Engineers & Scientists* (7th Ed.). Upper Saddle River, NJ: Prentice Hall

APPENDICES

A. Results Tables for the STRATCOM Example

The following tables were generated by the program. See Chapter 3 for details.

Table A.1: Probability of Task Success with All Asset Probabilities Ranging from 50 to 60 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.296892	0.72152	0.721443	0.666097
.1.	0.322834	0.665193	0.665757	0.666097
.2.	0.322115	0.665985	0.721443	0.600336
.3.	0.321938	0.665602	0.667346	0.666097
.9.	0.311457	0.72152	0.721443	0.596521
.10.	0.311309	0.72152	0.666421	0.599612
.11.	0.298807	0.72152	0.721443	0.602747
.1.2.	0.333372	0.598448	0.665757	0.600336
.1.3.	0.333259	0.597985	0.600837	0.666097
.1.9.	0.329242	0.665193	0.665757	0.596521
.1.10.	0.329179	0.665193	0.599725	0.599612
.1.11.	0.323549	0.665193	0.665757	0.602747
.2.3.	0.332802	0.598917	0.667346	0.600336
.2.9.	0.328652	0.665985	0.721443	0.517085
.2.10.	0.328632	0.665985	0.666421	0.520706
.2.11.	0.322821	0.665985	0.721443	0.524506
.3.9.	0.328565	0.665602	0.667346	0.596521
.3.10.	0.328524	0.665602	0.60165	0.599612
.3.11.	0.322654	0.665602	0.667346	0.602747
.9.10.	0.321143	0.72152	0.666421	0.516194
.9.11.	0.312422	0.72152	0.721443	0.519908
.10.11.	0.312329	0.72152	0.666421	0.523604
.1.2.3.	0.338755	0.517844	0.600837	0.600336
.1.2.9.	0.336824	0.598448	0.665757	0.517085
.1.2.10.	0.33682	0.598448	0.599725	0.520706
.1.2.11.	0.333642	0.598448	0.665757	0.524506
.1.3.9.	0.336753	0.597985	0.600837	0.596521
.1.3.10.	0.336745	0.597985	0.521993	0.599612
.1.3.11.	0.333537	0.597985	0.600837	0.602747
.1.9.10.	0.333757	0.665193	0.599725	0.516194
.1.9.11.	0.329603	0.665193	0.665757	0.519908
.1.10.11.	0.329566	0.665193	0.599725	0.523604
.2.3.9.	0.336372	0.598917	0.667346	0.517085
.2.3.10.	0.33638	0.598917	0.60165	0.520706
.2.3.11.	0.333077	0.598917	0.667346	0.524506
.2.9.10.	0.333296	0.665985	0.666421	0.42088
.2.9.11.	0.329015	0.665985	0.721443	0.425384
.2.10.11.	0.329009	0.665985	0.666421	0.429714
.3.9.10.	0.333248	0.665602	0.60165	0.516194

Table A.1: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.3.9.11.	0.328922	0.665602	0.667346	0.519908
.3.10.11.	0.328904	0.665602	0.60165	0.523604
.9.10.11.	0.321665	0.72152	0.666421	0.424275
.1.2.3.9.	0.340886	0.517844	0.600837	0.517085
.1.2.3.10.	0.340901	0.517844	0.521993	0.520706
.1.2.3.11.	0.338876	0.517844	0.600837	0.524506
.1.2.9.10.	0.339407	0.598448	0.599725	0.42088
.1.2.9.11.	0.336974	0.598448	0.665757	0.425384
.1.2.10.11.	0.336978	0.598448	0.599725	0.429714
.1.3.9.10.	0.339361	0.597985	0.521993	0.516194
.1.3.9.11.	0.336903	0.597985	0.600837	0.519908
.1.3.10.11.	0.336907	0.597985	0.521993	0.523604
.1.9.10.11.	0.333969	0.665193	0.599725	0.424275
.2.3.9.10.	0.339051	0.598917	0.60165	0.42088
.2.3.9.11.	0.336524	0.598917	0.667346	0.425384
.2.3.10.11.	0.336537	0.598917	0.60165	0.429714
.2.9.10.11.	0.333505	0.665985	0.666421	0.310846
.3.9.10.11.	0.33345	0.665602	0.60165	0.424275
.1.2.3.9.10.	0.342551	0.517844	0.521993	0.42088
.1.2.3.9.11.	0.340958	0.517844	0.600837	0.425384
.1.2.3.10.11.	0.340977	0.517844	0.521993	0.429714
.1.2.9.10.11.	0.339503	0.598448	0.599725	0.310846
.1.3.9.10.11.	0.339455	0.597985	0.521993	0.424275
.2.3.9.10.11.	0.339145	0.598917	0.60165	0.310846
.1.2.3.9.10.11.	0.3426	0.517844	0.521993	0.310846

Table A.2: Probability of Task Success with Non-assignable Asset Probabilities Ranging from 50 to 60 and Others from 90 to 100 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.298364	0.998584	0.998568	0.999698
.1.	0.343285	0.990023	0.990338	0.999698
.2.	0.343126	0.990091	0.998568	0.997737
.3.	0.34336	0.990036	0.989861	0.999698
.9.	0.333092	0.998584	0.998568	0.998032
.10.	0.333166	0.998584	0.990029	0.997915
.11.	0.301737	0.998584	0.998568	0.997879
.1.2.	0.348301	0.929055	0.990338	0.997737
.1.3.	0.348379	0.929656	0.932338	0.999698
.1.9.	0.34706	0.990023	0.990338	0.998032
.1.10.	0.347233	0.990023	0.93255	0.997915
.1.11.	0.343553	0.990023	0.990338	0.997879
.2.3.	0.348377	0.931899	0.989861	0.997737
.2.9.	0.346985	0.990091	0.998568	0.985035
.2.10.	0.347114	0.990091	0.990029	0.984454

Table A.2: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.11.	0.34343	0.990091	0.998568	0.984327
.3.9.	0.347103	0.990036	0.989861	0.998032
.3.10.	0.347166	0.990036	0.927544	0.997915
.3.11.	0.343618	0.990036	0.989861	0.997879
.9.10.	0.343278	0.998584	0.990029	0.986394
.9.11.	0.333504	0.998584	0.998568	0.986032
.10.11.	0.333874	0.998584	0.990029	0.986014
.1.2.3.	0.349675	0.513807	0.932338	0.997737
.1.2.9.	0.34937	0.929055	0.990338	0.985035
.1.2.10.	0.349517	0.929055	0.93255	0.984454
.1.2.11.	0.348362	0.929055	0.990338	0.984327
.1.3.9.	0.349423	0.929656	0.932338	0.998032
.1.3.10.	0.349506	0.929656	0.520756	0.997915
.1.3.11.	0.34843	0.929656	0.932338	0.997879
.1.9.10.	0.348923	0.990023	0.93255	0.986394
.1.9.11.	0.347128	0.990023	0.990338	0.986032
.1.10.11.	0.347345	0.990023	0.93255	0.986014
.2.3.9.	0.349426	0.931899	0.989861	0.985035
.2.3.10.	0.349488	0.931899	0.927544	0.984454
.2.3.11.	0.348427	0.931899	0.989861	0.984327
.2.9.10.	0.348861	0.990091	0.990029	0.897713
.2.9.11.	0.347068	0.990091	0.998568	0.894986
.2.10.11.	0.347244	0.990091	0.990029	0.896634
.3.9.10.	0.348859	0.990036	0.927544	0.986394
.3.9.11.	0.347168	0.990036	0.989861	0.986032
.3.10.11.	0.347264	0.990036	0.927544	0.986014
.9.10.11.	0.343422	0.998584	0.990029	0.907742
.1.2.3.9.	0.350149	0.513807	0.932338	0.985035
.1.2.3.10.	0.350223	0.513807	0.520756	0.984454
.1.2.3.11.	0.349691	0.513807	0.932338	0.984327
.1.2.9.10.	0.350094	0.929055	0.93255	0.897713
.1.2.9.11.	0.349388	0.929055	0.990338	0.894986
.1.2.10.11.	0.349553	0.929055	0.93255	0.896634
.1.3.9.10.	0.350081	0.929656	0.520756	0.986394
.1.3.9.11.	0.349442	0.929656	0.932338	0.986032
.1.3.10.11.	0.349534	0.929656	0.520756	0.986014
.1.9.10.11.	0.348958	0.990023	0.93255	0.907742
.2.3.9.10.	0.350072	0.931899	0.927544	0.897713
.2.3.9.11.	0.349448	0.931899	0.989861	0.894986
.2.3.10.11.	0.349518	0.931899	0.927544	0.896634
.2.9.10.11.	0.348901	0.990091	0.990029	0.311137
.3.9.10.11.	0.348893	0.990036	0.927544	0.907742
.1.2.3.9.10.	0.350525	0.513807	0.520756	0.897713
.1.2.3.9.11.	0.350156	0.513807	0.932338	0.894986
.1.2.3.10.11.	0.350234	0.513807	0.520756	0.896634
.1.2.9.10.11.	0.350108	0.929055	0.93255	0.311137
.1.3.9.10.11.	0.350092	0.929656	0.520756	0.907742

Table A.2: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.3.9.10.11.	0.350085	0.931899	0.927544	0.311137
.1.2.3.9.10.11.	0.350531	0.513807	0.520756	0.311137

Table A.3: Probability of Task Success with All Asset Probabilities Ranging from 60 to 70 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.620671	0.894499	0.895345	0.852976
.1.	0.629365	0.854906	0.854603	0.852976
.2.	0.629411	0.853794	0.895345	0.795813
.3.	0.629626	0.854329	0.855935	0.852976
.9.	0.626477	0.894499	0.895345	0.797128
.10.	0.626491	0.894499	0.855315	0.798959
.11.	0.62109	0.894499	0.895345	0.79753
.1.2.	0.632348	0.798912	0.854603	0.795813
.1.3.	0.632452	0.799683	0.799869	0.852976
.1.9.	0.631439	0.854906	0.854603	0.797128
.1.10.	0.631425	0.854906	0.798992	0.798959
.1.11.	0.629463	0.854906	0.854603	0.79753
.2.3.	0.632493	0.798125	0.855935	0.795813
.2.9.	0.631504	0.853794	0.895345	0.718176
.2.10.	0.631468	0.853794	0.855315	0.720696
.2.11.	0.629507	0.853794	0.895345	0.718844
.3.9.	0.631627	0.854329	0.855935	0.797128
.3.10.	0.631607	0.854329	0.800836	0.798959
.3.11.	0.629716	0.854329	0.855935	0.79753
.9.10.	0.629723	0.894499	0.855315	0.722606
.9.11.	0.626662	0.894499	0.895345	0.720651
.10.11.	0.626661	0.894499	0.855315	0.723215
.1.2.3.	0.633795	0.722377	0.799869	0.795813
.1.2.9.	0.633363	0.798912	0.854603	0.718176
.1.2.10.	0.633332	0.798912	0.798992	0.720696
.1.2.11.	0.632378	0.798912	0.854603	0.718844
.1.3.9.	0.633427	0.799683	0.799869	0.797128
.1.3.10.	0.633405	0.799683	0.723329	0.798959
.1.3.11.	0.63248	0.799683	0.799869	0.79753
.1.9.10.	0.632737	0.854906	0.798992	0.722606
.1.9.11.	0.63149	0.854906	0.854603	0.720651
.1.10.11.	0.631473	0.854906	0.798992	0.723215
.2.3.9.	0.633473	0.798125	0.855935	0.718176
.2.3.10.	0.633439	0.798125	0.800836	0.720696
.2.3.11.	0.632519	0.798125	0.855935	0.718844
.2.9.10.	0.632789	0.853794	0.855315	0.614515
.2.9.11.	0.63155	0.853794	0.895345	0.611981
.2.10.11.	0.631513	0.853794	0.855315	0.615514

Table A.3: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.3.9.10.	0.632872	0.854329	0.800836	0.722606
.3.9.11.	0.631671	0.854329	0.855935	0.720651
.3.10.11.	0.631649	0.854329	0.800836	0.723215
.9.10.11.	0.629805	0.894499	0.855315	0.618138
.1.2.3.9.	0.634353	0.722377	0.799869	0.718176
.1.2.3.10.	0.634323	0.722377	0.723329	0.720696
.1.2.3.11.	0.633805	0.722377	0.799869	0.718844
.1.2.9.10.	0.634022	0.798912	0.798992	0.614515
.1.2.9.11.	0.633381	0.798912	0.854603	0.611981
.1.2.10.11.	0.63335	0.798912	0.798992	0.615514
.1.3.9.10.	0.634069	0.799683	0.723329	0.722606
.1.3.9.11.	0.633444	0.799683	0.799869	0.720651
.1.3.10.11.	0.633421	0.799683	0.723329	0.723215
.1.9.10.11.	0.632765	0.854906	0.798992	0.618138
.2.3.9.10.	0.634106	0.798125	0.800836	0.614515
.2.3.9.11.	0.633488	0.798125	0.855935	0.611981
.2.3.10.11.	0.633454	0.798125	0.800836	0.615514
.2.9.10.11.	0.632815	0.853794	0.855315	0.469402
.3.9.10.11.	0.632895	0.854329	0.800836	0.618138
.1.2.3.9.10.	0.634722	0.722377	0.723329	0.614515
.1.2.3.9.11.	0.63436	0.722377	0.799869	0.611981
.1.2.3.10.11.	0.634331	0.722377	0.723329	0.615514
.1.2.9.10.11.	0.634034	0.798912	0.798992	0.469402
.1.3.9.10.11.	0.634079	0.799683	0.723329	0.618138
.2.3.9.10.11.	0.634116	0.798125	0.800836	0.469402
.1.2.3.9.10.11.	0.634727	0.722377	0.723329	0.469402

Table A.4: Probability of Task Success with Non-assignable Asset Probabilities Ranging from 60 to 70 and Others from 90 to 100 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.621644	0.999212	0.999217	0.999811
.1.	0.634161	0.994113	0.994319	0.999811
.2.	0.634115	0.994535	0.999217	0.998705
.3.	0.634235	0.994556	0.994773	0.999811
.9.	0.631667	0.999212	0.999217	0.99862
.10.	0.631533	0.999212	0.994097	0.99854
.11.	0.622218	0.999212	0.999217	0.998613
.1.2.	0.635406	0.959811	0.994319	0.998705
.1.3.	0.635471	0.95983	0.962229	0.999811
.1.9.	0.635145	0.994113	0.994319	0.99862
.1.10.	0.635072	0.994113	0.958926	0.99854
.1.11.	0.634201	0.994113	0.994319	0.998613
.2.3.	0.635473	0.961286	0.994773	0.998705
.2.9.	0.635102	0.994535	0.999217	0.990571

Table A.4: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.10.	0.635031	0.994535	0.994097	0.989837
.2.11.	0.634153	0.994535	0.999217	0.990348
.3.9.	0.635146	0.994556	0.994773	0.99862
.3.10.	0.635152	0.994556	0.959839	0.99854
.3.11.	0.634269	0.994556	0.994773	0.998613
.9.10.	0.634207	0.999212	0.994097	0.989446
.9.11.	0.631748	0.999212	0.999217	0.989855
.10.11.	0.631638	0.999212	0.994097	0.989723
.1.2.3.	0.635795	0.719452	0.962229	0.998705
.1.2.9.	0.635683	0.959811	0.994319	0.990571
.1.2.10.	0.635608	0.959811	0.958926	0.989837
.1.2.11.	0.635413	0.959811	0.994319	0.990348
.1.3.9.	0.635724	0.95983	0.962229	0.99862
.1.3.10.	0.635697	0.95983	0.722418	0.99854
.1.3.11.	0.635478	0.95983	0.962229	0.998613
.1.9.10.	0.635531	0.994113	0.958926	0.989446
.1.9.11.	0.635158	0.994113	0.994319	0.989855
.1.10.11.	0.635087	0.994113	0.958926	0.989723
.2.3.9.	0.635719	0.961286	0.994773	0.990571
.2.3.10.	0.635706	0.961286	0.959839	0.989837
.2.3.11.	0.635479	0.961286	0.994773	0.990348
.2.9.10.	0.635494	0.994535	0.994097	0.926653
.2.9.11.	0.635113	0.994535	0.999217	0.930126
.2.10.11.	0.635044	0.994535	0.994097	0.927636
.3.9.10.	0.635562	0.994556	0.959839	0.989446
.3.9.11.	0.635156	0.994556	0.994773	0.989855
.3.10.11.	0.635165	0.994556	0.959839	0.989723
.9.10.11.	0.634233	0.999212	0.994097	0.925618
.1.2.3.9.	0.635917	0.719452	0.962229	0.990571
.1.2.3.10.	0.63588	0.719452	0.722418	0.989837
.1.2.3.11.	0.635797	0.719452	0.962229	0.990348
.1.2.9.10.	0.635797	0.959811	0.958926	0.926653
.1.2.9.11.	0.635686	0.959811	0.994319	0.930126
.1.2.10.11.	0.635611	0.959811	0.958926	0.927636
.1.3.9.10.	0.635858	0.95983	0.722418	0.989446
.1.3.9.11.	0.635728	0.95983	0.962229	0.989855
.1.3.10.11.	0.635701	0.95983	0.722418	0.989723
.1.9.10.11.	0.635537	0.994113	0.958926	0.925618
.2.3.9.10.	0.635862	0.961286	0.959839	0.926653
.2.3.9.11.	0.635722	0.961286	0.994773	0.930126
.2.3.10.11.	0.635709	0.961286	0.959839	0.927636
.2.9.10.11.	0.635498	0.994535	0.994097	0.479653
.3.9.10.11.	0.635567	0.994556	0.959839	0.925618
.1.2.3.9.10.	0.635981	0.719452	0.722418	0.926653
.1.2.3.9.11.	0.635918	0.719452	0.962229	0.930126
.1.2.3.10.11.	0.635882	0.719452	0.722418	0.927636
.1.2.9.10.11.	0.635799	0.959811	0.958926	0.479653

Table A.4: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.1.3.9.10.11.	0.63586	0.95983	0.722418	0.925618
.2.3.9.10.11.	0.635863	0.961286	0.959839	0.479653
.1.2.3.9.10.11.	0.635982	0.719452	0.722418	0.479653

Table A.5: Probability of Task Success with Asset Probabilities Ranging from 70 to 80 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.898556	0.978016	0.978435	0.962577
.1.	0.899229	0.962215	0.96207	0.962577
.2.	0.899274	0.961966	0.978435	0.935385
.3.	0.899244	0.962136	0.962714	0.962577
.9.	0.89907	0.978016	0.978435	0.936047
.10.	0.899083	0.978016	0.962984	0.934574
.11.	0.898573	0.978016	0.978435	0.935188
.1.2.	0.899457	0.934673	0.96207	0.935385
.1.3.	0.899441	0.934864	0.934429	0.962577
.1.9.	0.899387	0.962215	0.96207	0.936047
.1.10.	0.899394	0.962215	0.934886	0.934574
.1.11.	0.899233	0.962215	0.96207	0.935188
.2.3.	0.899463	0.934495	0.962714	0.935385
.2.9.	0.899416	0.961966	0.978435	0.889647
.2.10.	0.899421	0.961966	0.962984	0.887041
.2.11.	0.899277	0.961966	0.978435	0.888083
.3.9.	0.899397	0.962136	0.962714	0.936047
.3.10.	0.899405	0.962136	0.935991	0.934574
.3.11.	0.899247	0.962136	0.962714	0.935188
.9.10.	0.899315	0.978016	0.962984	0.888176
.9.11.	0.899076	0.978016	0.978435	0.889181
.10.11.	0.899091	0.978016	0.962984	0.886697
.1.2.3.	0.89954	0.887393	0.934429	0.935385
.1.2.9.	0.899518	0.934673	0.96207	0.889647
.1.2.10.	0.899521	0.934673	0.934886	0.887041
.1.2.11.	0.899458	0.934673	0.96207	0.888083
.1.3.9.	0.899507	0.934864	0.934429	0.936047
.1.3.10.	0.899511	0.934864	0.887421	0.934574
.1.3.11.	0.899442	0.934864	0.934429	0.935188
.1.9.10.	0.899477	0.962215	0.934886	0.888176
.1.9.11.	0.899388	0.962215	0.96207	0.889181
.1.10.11.	0.899396	0.962215	0.934886	0.886697
.2.3.9.	0.899523	0.934495	0.962714	0.889647
.2.3.10.	0.899526	0.934495	0.935991	0.887041
.2.3.11.	0.899464	0.934495	0.962714	0.888083
.2.9.10.	0.899497	0.961966	0.962984	0.807048
.2.9.11.	0.899417	0.961966	0.978435	0.808758

Table A.5: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.10.11.	0.899423	0.961966	0.962984	0.804358
.3.9.10.	0.899486	0.962136	0.935991	0.888176
.3.9.11.	0.899399	0.962136	0.962714	0.889181
.3.10.11.	0.899406	0.962136	0.935991	0.886697
.9.10.11.	0.899317	0.978016	0.962984	0.806234
.1.2.3.9.	0.89957	0.887393	0.934429	0.889647
.1.2.3.10.	0.899572	0.887393	0.887421	0.887041
.1.2.3.11.	0.899541	0.887393	0.934429	0.888083
.1.2.9.10.	0.899557	0.934673	0.934886	0.807048
.1.2.9.11.	0.899518	0.934673	0.96207	0.808758
.1.2.10.11.	0.899522	0.934673	0.934886	0.804358
.1.3.9.10.	0.899549	0.934864	0.887421	0.888176
.1.3.9.11.	0.899507	0.934864	0.934429	0.889181
.1.3.10.11.	0.899512	0.934864	0.887421	0.886697
.1.9.10.11.	0.899478	0.962215	0.934886	0.806234
.2.3.9.10.	0.899561	0.934495	0.935991	0.807048
.2.3.9.11.	0.899523	0.934495	0.962714	0.808758
.2.3.10.11.	0.899527	0.934495	0.935991	0.804358
.2.9.10.11.	0.899497	0.961966	0.962984	0.66562
.3.9.10.11.	0.899487	0.962136	0.935991	0.806234
.1.2.3.9.10.	0.899591	0.887393	0.887421	0.807048
.1.2.3.9.11.	0.89957	0.887393	0.934429	0.808758
.1.2.3.10.11.	0.899573	0.887393	0.887421	0.804358
.1.2.9.10.11.	0.899557	0.934673	0.934886	0.66562
.1.3.9.10.11.	0.89955	0.934864	0.887421	0.806234
.2.3.9.10.11.	0.899561	0.934495	0.935991	0.66562
.1.2.3.9.10.11.	0.899591	0.887393	0.887421	0.66562

Table A.6: Probability of Task Success with Non-assignable Asset Probabilities Ranging from 70 to 80 and Others from 90 to 100 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.896642	0.999708	0.999732	0.999864
.1.	0.897645	0.997797	0.997806	0.999864
.2.	0.897623	0.997953	0.999732	0.999026
.3.	0.897654	0.997794	0.998228	0.999864
.9.	0.89747	0.999708	0.999732	0.999046
.10.	0.897484	0.999708	0.997872	0.999099
.11.	0.896664	0.999708	0.999732	0.999003
.1.2.	0.897754	0.984956	0.997806	0.999026
.1.3.	0.897772	0.983524	0.985513	0.999864
.1.9.	0.897738	0.997797	0.997806	0.999046
.1.10.	0.897737	0.997797	0.98333	0.999099
.1.11.	0.897647	0.997797	0.997806	0.999003
.2.3.	0.897758	0.984908	0.998228	0.999026

Table A.6: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.9.	0.897718	0.997953	0.999732	0.993222
.2.10.	0.897724	0.997953	0.997872	0.993527
.2.11.	0.897625	0.997953	0.999732	0.992872
.3.9.	0.897757	0.997794	0.998228	0.999046
.3.10.	0.89775	0.997794	0.98541	0.999099
.3.11.	0.897656	0.997794	0.998228	0.999003
.9.10.	0.897688	0.999708	0.997872	0.99373
.9.11.	0.897477	0.999708	0.999732	0.992752
.10.11.	0.89749	0.999708	0.997872	0.9935
.1.2.3.	0.897802	0.889201	0.985513	0.999026
.1.2.9.	0.897786	0.984956	0.997806	0.993222
.1.2.10.	0.897786	0.984956	0.98333	0.993527
.1.2.11.	0.897754	0.984956	0.997806	0.992872
.1.3.9.	0.897808	0.983524	0.985513	0.999046
.1.3.10.	0.897801	0.983524	0.887175	0.999099
.1.3.11.	0.897773	0.983524	0.985513	0.999003
.1.9.10.	0.897781	0.997797	0.98333	0.99373
.1.9.11.	0.897739	0.997797	0.997806	0.992752
.1.10.11.	0.897737	0.997797	0.98333	0.9935
.2.3.9.	0.897796	0.984908	0.998228	0.993222
.2.3.10.	0.897791	0.984908	0.98541	0.993527
.2.3.11.	0.897758	0.984908	0.998228	0.992872
.2.9.10.	0.897768	0.997953	0.997872	0.955546
.2.9.11.	0.897719	0.997953	0.999732	0.948631
.2.10.11.	0.897725	0.997953	0.997872	0.953735
.3.9.10.	0.897798	0.997794	0.98541	0.99373
.3.9.11.	0.897757	0.997794	0.998228	0.992752
.3.10.11.	0.897751	0.997794	0.98541	0.9935
.9.10.11.	0.89769	0.999708	0.997872	0.95334
.1.2.3.9.	0.897824	0.889201	0.985513	0.993222
.1.2.3.10.	0.897818	0.889201	0.887175	0.993527
.1.2.3.11.	0.897802	0.889201	0.985513	0.992872
.1.2.9.10.	0.897808	0.984956	0.98333	0.955546
.1.2.9.11.	0.897786	0.984956	0.997806	0.948631
.1.2.10.11.	0.897786	0.984956	0.98333	0.953735
.1.3.9.10.	0.897826	0.983524	0.887175	0.99373
.1.3.9.11.	0.897808	0.983524	0.985513	0.992752
.1.3.10.11.	0.897801	0.983524	0.887175	0.9935
.1.9.10.11.	0.897781	0.997797	0.98333	0.95334
.2.3.9.10.	0.897816	0.984908	0.98541	0.955546
.2.3.9.11.	0.897796	0.984908	0.998228	0.948631
.2.3.10.11.	0.897791	0.984908	0.98541	0.953735
.2.9.10.11.	0.897769	0.997953	0.997872	0.672326
.3.9.10.11.	0.897798	0.997794	0.98541	0.95334
.1.2.3.9.10.	0.897836	0.889201	0.887175	0.955546
.1.2.3.9.11.	0.897824	0.889201	0.985513	0.948631
.1.2.3.10.11.	0.897818	0.889201	0.887175	0.953735

Table A.6: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.1.2.9.10.11.	0.897808	0.984956	0.98333	0.672326
.1.3.9.10.11.	0.897826	0.983524	0.887175	0.95334
.2.3.9.10.11.	0.897816	0.984908	0.98541	0.672326
.1.2.3.9.10.11.	0.897836	0.889201	0.887175	0.672326

Table A.7: Probability of Task Success with Asset Probabilities Ranging from 80 to 90 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.99413	0.998665	0.998811	0.996789
.1.	0.994135	0.996415	0.996801	0.996789
.2.	0.994135	0.996575	0.998811	0.991528
.3.	0.994135	0.996608	0.996931	0.996789
.9.	0.994135	0.998665	0.998811	0.991719
.10.	0.994135	0.998665	0.996923	0.991602
.11.	0.99413	0.998665	0.998811	0.99182
.1.2.	0.994137	0.990817	0.996801	0.991528
.1.3.	0.994137	0.990896	0.991743	0.996789
.1.9.	0.994137	0.996415	0.996801	0.991719
.1.10.	0.994137	0.996415	0.991743	0.991602
.1.11.	0.994135	0.996415	0.996801	0.99182
.2.3.	0.994137	0.991336	0.996931	0.991528
.2.9.	0.994137	0.996575	0.998811	0.978186
.2.10.	0.994137	0.996575	0.996923	0.977849
.2.11.	0.994136	0.996575	0.998811	0.978426
.3.9.	0.994137	0.996608	0.996931	0.991719
.3.10.	0.994137	0.996608	0.992053	0.991602
.3.11.	0.994135	0.996608	0.996931	0.99182
.9.10.	0.994137	0.998665	0.996923	0.978308
.9.11.	0.994135	0.998665	0.998811	0.978932
.10.11.	0.994135	0.998665	0.996923	0.978601
.1.2.3.	0.994138	0.976771	0.991743	0.991528
.1.2.9.	0.994138	0.990817	0.996801	0.978186
.1.2.10.	0.994138	0.990817	0.991743	0.977849
.1.2.11.	0.994137	0.990817	0.996801	0.978426
.1.3.9.	0.994138	0.990896	0.991743	0.991719
.1.3.10.	0.994138	0.990896	0.978682	0.991602
.1.3.11.	0.994137	0.990896	0.991743	0.99182
.1.9.10.	0.994138	0.996415	0.991743	0.978308
.1.9.11.	0.994137	0.996415	0.996801	0.978932
.1.10.11.	0.994137	0.996415	0.991743	0.978601
.2.3.9.	0.994138	0.991336	0.996931	0.978186
.2.3.10.	0.994138	0.991336	0.992053	0.977849
.2.3.11.	0.994137	0.991336	0.996931	0.978426
.2.9.10.	0.994138	0.996575	0.996923	0.942886

Table A.7: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.9.11.	0.994137	0.996575	0.998811	0.944532
.2.10.11.	0.994137	0.996575	0.996923	0.943597
.3.9.10.	0.994138	0.996608	0.992053	0.978308
.3.9.11.	0.994137	0.996608	0.996931	0.978932
.3.10.11.	0.994137	0.996608	0.992053	0.978601
.9.10.11.	0.994137	0.998665	0.996923	0.944819
.1.2.3.9.	0.994138	0.976771	0.991743	0.978186
.1.2.3.10.	0.994138	0.976771	0.978682	0.977849
.1.2.3.11.	0.994138	0.976771	0.991743	0.978426
.1.2.9.10.	0.994138	0.990817	0.991743	0.942886
.1.2.9.11.	0.994138	0.990817	0.996801	0.944532
.1.2.10.11.	0.994138	0.990817	0.991743	0.943597
.1.3.9.10.	0.994138	0.990896	0.978682	0.978308
.1.3.9.11.	0.994138	0.990896	0.991743	0.978932
.1.3.10.11.	0.994138	0.990896	0.978682	0.978601
.1.9.10.11.	0.994138	0.996415	0.991743	0.944819
.2.3.9.10.	0.994138	0.991336	0.992053	0.942886
.2.3.9.11.	0.994138	0.991336	0.996931	0.944532
.2.3.10.11.	0.994138	0.991336	0.992053	0.943597
.2.9.10.11.	0.994138	0.996575	0.996923	0.854803
.3.9.10.11.	0.994138	0.996608	0.992053	0.944819
.1.2.3.9.10.	0.994138	0.976771	0.978682	0.942886
.1.2.3.9.11.	0.994138	0.976771	0.991743	0.944532
.1.2.3.10.11.	0.994138	0.976771	0.978682	0.943597
.1.2.9.10.11.	0.994138	0.990817	0.991743	0.854803
.1.3.9.10.11.	0.994138	0.990896	0.978682	0.944819
.2.3.9.10.11.	0.994138	0.991336	0.992053	0.854803
.1.2.3.9.10.11.	0.994138	0.976771	0.978682	0.854803

Table A.8: Probability of Task Success with Non-assignable Asset Probabilities Ranging from 80 to 90 and Others from 90 to 100 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.993982	0.999931	0.999938	0.999948
.1.	0.993989	0.999532	0.999614	0.999948
.2.	0.99399	0.999562	0.999938	0.999627
.3.	0.993989	0.999503	0.999557	0.999948
.9.	0.993989	0.999931	0.999938	0.999602
.10.	0.993989	0.999931	0.999551	0.999608
.11.	0.993982	0.999931	0.999938	0.99964
.1.2.	0.993991	0.997049	0.999614	0.999627
.1.3.	0.993991	0.99654	0.997224	0.999948
.1.9.	0.993991	0.999532	0.999614	0.999602
.1.10.	0.993991	0.999532	0.997176	0.999608
.1.11.	0.993989	0.999532	0.999614	0.99964

Table A.8: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.3.	0.993991	0.996867	0.999557	0.999627
.2.9.	0.993991	0.999562	0.999938	0.997202
.2.10.	0.993991	0.999562	0.999551	0.997207
.2.11.	0.99399	0.999562	0.999938	0.997417
.3.9.	0.99399	0.999503	0.999557	0.999602
.3.10.	0.99399	0.999503	0.996728	0.999608
.3.11.	0.993989	0.999503	0.999557	0.99964
.9.10.	0.993991	0.999931	0.999551	0.997066
.9.11.	0.993989	0.999931	0.999938	0.997104
.10.11.	0.993989	0.999931	0.999551	0.997311
.1.2.3.	0.993991	0.978373	0.997224	0.999627
.1.2.9.	0.993992	0.997049	0.999614	0.997202
.1.2.10.	0.993992	0.997049	0.997176	0.997207
.1.2.11.	0.993991	0.997049	0.999614	0.997417
.1.3.9.	0.993991	0.99654	0.997224	0.999602
.1.3.10.	0.993991	0.99654	0.979266	0.999608
.1.3.11.	0.993991	0.99654	0.997224	0.99964
.1.9.10.	0.993991	0.999532	0.997176	0.997066
.1.9.11.	0.993991	0.999532	0.999614	0.997104
.1.10.11.	0.993991	0.999532	0.997176	0.997311
.2.3.9.	0.993991	0.996867	0.999557	0.997202
.2.3.10.	0.993991	0.996867	0.996728	0.997207
.2.3.11.	0.993991	0.996867	0.999557	0.997417
.2.9.10.	0.993991	0.999562	0.999551	0.979391
.2.9.11.	0.993991	0.999562	0.999938	0.979591
.2.10.11.	0.993991	0.999562	0.999551	0.980838
.3.9.10.	0.993991	0.999503	0.996728	0.997066
.3.9.11.	0.99399	0.999503	0.999557	0.997104
.3.10.11.	0.99399	0.999503	0.996728	0.997311
.9.10.11.	0.993991	0.999931	0.999551	0.97886
.1.2.3.9.	0.993992	0.978373	0.997224	0.997202
.1.2.3.10.	0.993992	0.978373	0.979266	0.997207
.1.2.3.11.	0.993991	0.978373	0.997224	0.997417
.1.2.9.10.	0.993992	0.997049	0.997176	0.979391
.1.2.9.11.	0.993992	0.997049	0.999614	0.979591
.1.2.10.11.	0.993992	0.997049	0.997176	0.980838
.1.3.9.10.	0.993992	0.99654	0.979266	0.997066
.1.3.9.11.	0.993991	0.99654	0.997224	0.997104
.1.3.10.11.	0.993991	0.99654	0.979266	0.997311
.1.9.10.11.	0.993991	0.999532	0.997176	0.97886
.2.3.9.10.	0.993992	0.996867	0.996728	0.979391
.2.3.9.11.	0.993991	0.996867	0.999557	0.979591
.2.3.10.11.	0.993991	0.996867	0.996728	0.980838
.2.9.10.11.	0.993991	0.999562	0.999551	0.85122
.3.9.10.11.	0.993991	0.999503	0.996728	0.97886
.1.2.3.9.10.	0.993992	0.978373	0.979266	0.979391
.1.2.3.9.11.	0.993992	0.978373	0.997224	0.979591

Table A.8: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.1.2.3.10.11.	0.993992	0.978373	0.979266	0.980838
.1.2.9.10.11.	0.993992	0.997049	0.997176	0.85122
.1.3.9.10.11.	0.993992	0.99654	0.979266	0.97886
.2.3.9.10.11.	0.993992	0.996867	0.996728	0.85122
.1.2.3.9.10.11.	0.993992	0.978373	0.979266	0.85122

Table A.9: Probability of Task Success with Asset Probabilities Ranging from 90 to 100 Percent

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
None	0.999994	0.999999	0.999999	0.999991
.1.	0.999994	0.999991	0.99999	0.999991
.2.	0.999994	0.999992	0.999999	0.999943
.3.	0.999994	0.999992	0.99999	0.999991
.9.	0.999994	0.999999	0.999999	0.999941
.10.	0.999994	0.999999	0.999991	0.999941
.11.	0.999994	0.999999	0.999999	0.999944
.1.2.	0.999994	0.999946	0.99999	0.999943
.1.3.	0.999994	0.999939	0.999929	0.999991
.1.9.	0.999994	0.999991	0.99999	0.999941
.1.10.	0.999994	0.999991	0.99994	0.999941
.1.11.	0.999994	0.999991	0.99999	0.999944
.2.3.	0.999994	0.999947	0.99999	0.999943
.2.9.	0.999994	0.999992	0.999999	0.999612
.2.10.	0.999994	0.999992	0.999991	0.9996
.2.11.	0.999994	0.999992	0.999999	0.99962
.3.9.	0.999994	0.999992	0.99999	0.999941
.3.10.	0.999994	0.999992	0.999937	0.999941
.3.11.	0.999994	0.999992	0.99999	0.999944
.9.10.	0.999994	0.999999	0.999991	0.999592
.9.11.	0.999994	0.999999	0.999999	0.99962
.10.11.	0.999994	0.999999	0.999991	0.999627
.1.2.3.	0.999994	0.999618	0.999929	0.999943
.1.2.9.	0.999994	0.999946	0.99999	0.999612
.1.2.10.	0.999994	0.999946	0.99994	0.9996
.1.2.11.	0.999994	0.999946	0.99999	0.99962
.1.3.9.	0.999994	0.999939	0.999929	0.999941
.1.3.10.	0.999994	0.999939	0.999563	0.999941
.1.3.11.	0.999994	0.999939	0.999929	0.999944
.1.9.10.	0.999994	0.999991	0.99994	0.999592
.1.9.11.	0.999994	0.999991	0.99999	0.99962
.1.10.11.	0.999994	0.999991	0.99994	0.999627
.2.3.9.	0.999994	0.999947	0.99999	0.999612
.2.3.10.	0.999994	0.999947	0.999937	0.9996
.2.3.11.	0.999994	0.999947	0.99999	0.99962

Table A.9: Continued

Assets Assigned to Task 1	TASK1	TASK2	TASK3	TASK4
.2.9.10.	0.999994	0.999992	0.999991	0.997197
.2.9.11.	0.999994	0.999992	0.999999	0.997396
.2.10.11.	0.999994	0.999992	0.999991	0.997373
.3.9.10.	0.999994	0.999992	0.999937	0.999592
.3.9.11.	0.999994	0.999992	0.99999	0.99962
.3.10.11.	0.999994	0.999992	0.999937	0.999627
.9.10.11.	0.999994	0.999999	0.999991	0.997413
.1.2.3.9.	0.999994	0.999618	0.999929	0.999612
.1.2.3.10.	0.999994	0.999618	0.999563	0.9996
.1.2.3.11.	0.999994	0.999618	0.999929	0.99962
.1.2.9.10.	0.999994	0.999946	0.99994	0.997197
.1.2.9.11.	0.999994	0.999946	0.99999	0.997396
.1.2.10.11.	0.999994	0.999946	0.99994	0.997373
.1.3.9.10.	0.999994	0.999939	0.999563	0.999592
.1.3.9.11.	0.999994	0.999939	0.999929	0.99962
.1.3.10.11.	0.999994	0.999939	0.999563	0.999627
.1.9.10.11.	0.999994	0.999991	0.99994	0.997413
.2.3.9.10.	0.999994	0.999947	0.999937	0.997197
.2.3.9.11.	0.999994	0.999947	0.99999	0.997396
.2.3.10.11.	0.999994	0.999947	0.999937	0.997373
.2.9.10.11.	0.999994	0.999992	0.999991	0.981494
.3.9.10.11.	0.999994	0.999992	0.999937	0.997413
.1.2.3.9.10.	0.999994	0.999618	0.999563	0.997197
.1.2.3.9.11.	0.999994	0.999618	0.999929	0.997396
.1.2.3.10.11.	0.999994	0.999618	0.999563	0.997373
.1.2.9.10.11.	0.999994	0.999946	0.99994	0.981494
.1.3.9.10.11.	0.999994	0.999939	0.999563	0.997413
.2.3.9.10.11.	0.999994	0.999947	0.999937	0.981494
.1.2.3.9.10.11.	0.999994	0.999618	0.999563	0.981494

B. The Visual Basic Code

B.1. MainScrean.vb

```
Module MainScreen

  Sub Main()
    Dim repeat As Boolean = False

    Console.WriteLine("GSMOpt takes a network with assignable assets and calculates")
    Console.WriteLine("the end-stage probabilities of success for each network allocation." & vbCrLf)

    While True
      'Dimension all data arrays to be used to store information
      Dim msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
            New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
            New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
            New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
            New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))
```

```
  'This dictionary contains the entire task 1 network structure information.
Dim netConnect As Dictionary(Of String, Dictionary(Of String, String)) = _
      New Dictionary(Of String, Dictionary(Of String, String))

Dim assignable As String()
Dim networkAllocations As String()

'Initialize the netConnect dictionary which will contain the asset connection information for
'task 1.
Dim msDefStages As String() = New String() {"Stg3", "Stg4", "Stg5", "Stg6", "Stg7", "Stg8", _
                                  "Stg9", "Stg10"}
For Each stage As String In msDefStages
  netConnect(stage) = New Dictionary(Of String, String)
Next

'Initialize the data dictionaries.
InitDictionaries(msDef, spCol, intCol, topSec)

Console.WriteLine("Please select an option:")
Console.WriteLine("Enter ""n"" to load a new network and its connection information.")
Console.WriteLine("Enter ""e"" to load an existing network and its probabilities.")
Console.WriteLine("Enter ""r"" to perform network replications, and construct average")
Console.WriteLine("end-stage probabilities.")
Console.WriteLine("Enter ""q"" to quit.")
If repeat Then
  Console.WriteLine("Note: Probability of success files for each stage of task 1 generated")
  Console.WriteLine("for the last network will be overwritten.")
End If
Dim myOption As String = Console.ReadLine()
```

```
Console.WriteLine()

  Select Case myOption

    Case "n", "e"
      If myOption = "n" Then
        'Get network structure from a *.csv file which is located in the
        '...\GSMOpt\bin\Release project directory.
        Console.WriteLine("Please enter the name of the .csv file that contains the network connections ")
        Console.Write("(file must be located in ...\GSMOpt\bin\Release): ")
        Dim networkCSVFile As String = Console.ReadLine()

        Console.WriteLine()
        Console.WriteLine _
          ("Please enter the lower bound for the probability of success of an asset")
        Console.Write("in Stage 2 of Task 1 as a number between 0 and 100 (e.g. 95): ")
        Dim lowerBnd2 As Double = Console.ReadLine()
        Console.WriteLine _
          ("Please enter the upper bound for the probability of success of an asset")
        Console.WriteLine("in Stage 2 of Task 1 as a number between 0 and 100")
        Console.Write("(Must be greater than the lower bound): ")
        Dim upperBnd2 As Double = Console.ReadLine()

        Console.WriteLine(vbCrLf & "Probability of success for every other asset:")
        Console.WriteLine _
          ("Please enter the lower bound for the probability of success of an asset")
        Console.Write("as a number between 0 and 100 (e.g. 95): ")
        Dim lowerBnd As Double = Console.ReadLine()
```

```
    Console.WriteLine _
       ("Please enter the upper bound for the probability of success of an asset")
    Console.Write _
       ("as a number between 0 and 100 (Must be greater than the lower bound): ")
    Dim upperBnd As Double = Console.ReadLine()

    Console.WriteLine()
    Console.WriteLine _
       ("The network will now be imported, and the necessary probabilities")
    Console.WriteLine _
       ("for every asset at every stage of every task will be randomly generated.")
    Console.WriteLine("Please wait ...." & vbCrLf)

    'Takes care of importing the network connections, and generating probabilities
    'where necessary.
    InputHandling.ImportNetStruct(networkCSVFile, (lowerBnd2 / 100), (upperBnd2 / 100), (lowerBnd / 100), _
                        (upperBnd / 100), msDef, spCol, intCol, topSec, netConnect)

    'Saves the randomly generated probabilities for future reference.
    InputHandling.SaveProbabilities(msDef, spCol, intCol, topSec, netConnect)

Else
    'Imports the network structure and probabilities from a .csv file.
    Console.WriteLine("Please enter the name of the .csv file that contains the probability")
    Console.WriteLine("information (file must be located in ...\GSMOpt\bin\Release, and be")
    Console.Write("formatted correctly): ")
    Dim probCSVFile As String = Console.ReadLine()

    'Takes care of importing the network structure and probabilities.
```

```
    InputHandling.ImportNetStructAndProbs(probCSVFile, msDef, spCol, intCol, topSec, netConnect)
End If

Console.WriteLine(vbCrLf & "Please enter the name of the .csv file you want to save the end-stage")
Console.Write("probabilities for each network allocation in: ")
Dim endStageFile As String = Console.ReadLine()

Console.WriteLine(vbCrLf & "Network allocations are being iterated through, and")
Console.WriteLine("probabilities calculated for each network allocation.")
Console.WriteLine()

'Determine what assets in the network are assignable.
assignable = GenSystem.FindAssignable(msDef, spCol, intCol, topSec)

'Generate a list of the different combinations of assignable assets allocated to work on task 1.
networkAllocations = GenSystem.GenCombinations("No assets on task 1.", assignable)

'Optimize each network allocation. Each call to GenNetwork.NewNetwork stores the appropriate
'probabilities that have been calculated in the file specified.
Dim endStageArray As Double()() = New Double(networkAllocations.Length - 1)() {}

'The case where no assignable assets are allocated to task 1 is a special case.
endStageArray(0) = GenNetwork.NewNetwork(New String() {}, assignable, msDef, spCol, intCol, _
                   topSec, netConnect)

'Iterate over all non-empty allocations in the original network
For i As Integer = 1 To networkAllocations.Length - 1
    endStageArray(i) = GenNetwork.NewNetwork(Split(networkAllocations(i).Trim("."), "."), _
                       assignable, msDef, spCol, intCol, topSec, netConnect)
```

```
    Next

    'Export the probabilities to "endStageFile.csv".
    ExportEndStageProbs(endStageFile, networkAllocations, endStageArray, msDef("Stg10"))

    Console.WriteLine(vbCrLf & vbCrLf)

Case "r"
    'Get network structure from a *.csv file which is located in the ...\GSMOpt\bin\Release
    'project directory.
    Console.WriteLine("Please enter the name of the .csv file that contains the network connections ")
    Console.Write("(file must be located in ...\GSMOpt\bin\Release): ")
    Dim networkCSVFile As String = Console.ReadLine()

    Console.WriteLine()
    Console.WriteLine("Please enter the lower bound for the probability of success of an asset")
    Console.Write("in Stage 2 of Task 1 as a number between 0 and 100 (e.g. 95): ")
    Dim lowerBnd2 As Double = Console.ReadLine()
    Console.WriteLine("Please enter the upper bound for the probability of success of an asset")
    Console.WriteLine("in Stage 2 of Task 1 as a number between 0 and 100")
    Console.Write("(Must be greater than the lower bound): ")
    Dim upperBnd2 As Double = Console.ReadLine()

    Console.WriteLine(vbCrLf & "Probability of success for every other asset:")
    Console.WriteLine("Please enter the lower bound for the probability of success of an asset")
    Console.Write("as a number between 0 and 100 (e.g. 95): ")
    Dim lowerBnd As Double = Console.ReadLine()
    Console.WriteLine("Please enter the upper bound for the probability of success of an asset")
    Console.Write("as a number between 0 and 100 (Must be greater than the lower bound): ")
```

```
Dim upperBnd As Double = Console.ReadLine()

Console.WriteLine()
Console.Write("Enter the number of replications you wish to perform: ")
Dim reps As Integer = Console.ReadLine()

Console.WriteLine()
Console.WriteLine("The network will now be imported, and the necessary probabilities")
Console.WriteLine("for every asset at every stage of every task will be randomly generated.")
Console.WriteLine("Please wait ...." & vbCrLf)

'Takes care of importing the network connections, and generating probabilities where necessary.
InputHandling.ImportNetStruct(networkCSVFile, (lowerBnd2 / 100), (upperBnd2 / 100), (lowerBnd / 100), _
                              (upperBnd / 100), msDef, spCol, intCol, topSec, netConnect)

Console.WriteLine(vbCrLf & "Please enter the name of the .csv file you want to save the average end-stage")
Console.Write("probabilities for each network allocation in: ")
Dim avgStageFile As String = Console.ReadLine()

Console.WriteLine(vbCrLf & "Network allocations are being iterated through, and")
Console.WriteLine("probabilities calculated for each network allocation.")
Console.WriteLine()

'Determine what assets in the network are assignable.
assignable = GenSystem.FindAssignable(msDef, spCol, intCol, topSec)

'Generate a list of the different combinations of assignable assets allocated to work on task 1.
networkAllocations = GenSystem.GenCombinations("Assigned to task 1", assignable)
```

```
    Dim repsArray As Double(,)() = New Double(networkAllocations.Length - 1, reps - 1)() {}

    For j As Integer = 0 To reps - 1
        Console.WriteLine("Replication number " & (j + 1) & " being computed...")

        'The case where no assignable assets are allocated to task 1 is a special case.
        repsArray(0, j) = GenNetwork.NewNetwork(New String() {}, assignable, msDef, spCol, intCol, topSec, _
                netConnect)

        'Iterate over all non-empty allocations in the original network
        For i As Integer = 1 To networkAllocations.Length - 1
            repsArray(i, j) = GenNetwork.NewNetwork(Split(networkAllocations(i).Trim("."), "."), assignable, _
                msDef, spCol, intCol, topSec, netConnect)
        Next

        Console.WriteLine("Replication number " & (j + 1) & " finished.")

        'Randomly generate new probabilities for the entire network so that the next replication can start afresh.
        InputHandling.RegenProbs((lowerBnd2 / 100), (upperBnd2 / 100), (lowerBnd / 100), (upperBnd / 100), _
                        msDef, spCol, intCol, topSec)
    Next

    'Write the average end-stage probabilities to "avgStageFile.csv".
    ExportEndStageProbs(avgStageFile, networkAllocations, CalcAverages(reps, msDef("Stg10").Count, _
                                                repsArray), msDef("Stg10"))

    Console.WriteLine(vbCrLf & vbCrLf)

Case Else
```

```
            Exit While 'if "q" or any other option that is not "n" or "e" is entered quit
        End Select

      repeat = True
    End While
End Sub


'Initializes the data arrays used to store all the information. This initialization assumes that there
'are four task types, 1) Missile Defense (msDef), 2) Space Collect (spCol), 3) Intelligence Collect (intCol),
'4) Classified (topSec). In addition, it is assumed that missile defense has stage 2-10, and all other task
'types have stages 2, 4, and 5.

Sub InitDictionaries(ByRef task1 As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                     ByRef task2 As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                     ByRef task3 As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                     ByRef task4 As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

    Dim msDefStages As String() = New String() {"Stg2", "Stg3", "Stg4", "Stg5", "Stg6", "Stg7", "Stg8", _
                                                "Stg9", "Stg10"}
    Dim otherTskStages As String() = New String() {"Stg2", "Stg4", "Stg5"}


    For Each stage As String In msDefStages
      task1(stage) = New Dictionary(Of String, Dictionary(Of String, Double))
    Next

    For Each stage As String In otherTskStages
      task2(stage) = New Dictionary(Of String, Dictionary(Of String, Double))
```

```
    task3(stage) = New Dictionary(Of String, Dictionary(Of String, Double))
    task4(stage) = New Dictionary(Of String, Dictionary(Of String, Double))
  Next
End Sub


'Writes the end stage probabilities for a given network to the specified file.

 Private Sub ExportEndStageProbs(ByVal fileName As String, ByVal networkAllocations As String(), _
                                 ByVal probArray As Double()(), ByVal task1 As Dictionary(Of String,  _
                                                                 Dictionary(Of String, Double)))

  Using sw As StreamWriter = New StreamWriter(fileName & ".csv")
    sw.Write("NETWORK ALLOCATION,")

    For Each asset As String In task1.Keys
      sw.Write(asset & ",")
    Next

    sw.WriteLine("TASK2,TASK3,TASK4")

    For i As Integer = 0 To probArray.GetUpperBound(0)
      sw.Write(networkAllocations(i) & ",")

      For j As Integer = 0 To probArray(i).GetUpperBound(0) - 1
        sw.Write(probArray(i)(j) & ",")
      Next
      sw.WriteLine(probArray(i)(probArray(i).GetUpperBound(0)))
    Next
```

```
    End Using
  End Sub


  'Calculates the average probabilities when replications have been performed.

  Private Function CalcAverages(ByVal reps As Integer, ByVal numTask1Assets As Integer, ByVal repsArray As
Double(,)())
    Dim avgArray As Double()() = New Double(repsArray.GetUpperBound(0))() {}

    For i As Integer = 0 To repsArray.GetUpperBound(0)
      avgArray(i) = New Double(numTask1Assets + 3 - 1) {}

      For j As Integer = 0 To repsArray.GetUpperBound(1)
        For k As Integer = 0 To repsArray(i, j).GetUpperBound(0)
          avgArray(i)(k) += repsArray(i, j)(k)
        Next
      Next
    Next

    For i As Integer = 0 To avgArray.GetUpperBound(0)
      For j As Integer = 0 To avgArray(i).GetUpperBound(0)
        avgArray(i)(j) /= reps
      Next
    Next

    Return avgArray
  End Function
```

```
End Module
```

B.2. InputHandling.vb

```
Public Class InputHandling
  'ImportConnections reads in the network connections for a given system. The code works iff
  'the file it is reading in is formatted correctly, since it uses the structure of the file
  'as well as several keywords to read through the file correctly.
  'Specifically in reading in tasks 2, 3, and 4, the code takes advantage of the fact that
  'the tasks are linear in nature.

   Public Shared Sub ImportNetStruct(ByVal networkCSVFile As String, ByVal lowerBnd2 As Double, _
                ByVal upperBnd2 As Double, ByVal lowerBnd As Double, ByVal upperBnd As Double, _
                ByRef msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef netConnect As Dictionary(Of String, Dictionary(Of String, String)))

      '*NOTE* A *.csv file named "networkCSVFile.csv" that contains the network connections for the system has to be
      'in the ...\GSMOpt\bin\Release project directory.
      'This file has a special format, and if the data is not in this format, the following code will not work.

      'Create an instance of StreamReader to read from a file.
      Using sr As StreamReader = New StreamReader(networkCSVFile & ".csv")
         Dim nextLine As String
         Dim taskNotDone As Boolean = True
```

```
Dim stageNotDone As Boolean = True
Dim range As Double = upperBnd - lowerBnd
Dim randNum As New Random()

sr.ReadLine() 'read in "TASK1"
sr.ReadLine() 'read in "Stage2"
nextLine = sr.ReadLine() 'read in the first asset in "Stage2"
While stageNotDone
   'format nextLine so that it contains only the name of the asset
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If

   Dim myArray As String() = GenSystem.GenCombinations(nextLine, New String() {})

   msDef("Stg2")(myArray(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray
      msDef("Stg2")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd2, _
                                          (upperBnd2 - lowerBnd2))
   Next

   nextLine = sr.ReadLine
   If nextLine.IndexOf("endStage") <> -1 Then
      stageNotDone = False
   End If
End While
```

```
stageNotDone = True
sr.ReadLine() 'read in "Stage3"
nextLine = sr.ReadLine() 'read in the first asset in "Stage3"

While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If

   Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                nextLine.Substring(nextLine.IndexOf(",") + 1).Split(",")) 'gets the assets connected to the current asset

   'Place the combination containing all the assets connected to the current asset
   'in the dictionary.
   netConnect("Stg3").Item(myArray(0)) = myArray(myArray.Length - 1)

   msDef("Stg3")(myArray(0)) = New Dictionary(Of String, Double)

   Dim index As Integer = 0

   For Each combo As String In myArray
      Select Case combo.Trim(".").Split(".").Length
         Case 1
            msDef("Stg3")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
         Case Else
            Dim assetsInComb As String() = combo.Trim(".").Split(".")
```

```
            Dim maxProb As Double = 0

            For Each asst As String In assetsInComb
               If msDef("Stg3")(myArray(0)).Item("." & asst & ".") > maxProb Then
                  maxProb = msDef("Stg3")(myArray(0)).Item("." & asst & ".")
               End If
            Next

            msDef("Stg3")(myArray(0)).Item(combo) = maxProb
      End Select

      index += 1

   Next

   msDef("Stg3")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

   nextLine = sr.ReadLine
   If nextLine.IndexOf("endStage") <> -1 Then
      stageNotDone = False
   End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage4"
nextLine = sr.ReadLine() 'read in the first asset in "Stage4"

While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
```

```
   nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
Else
   nextLine = nextLine.Trim(",")
End If

'Retrieves the assets connected to the current asset
Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
            nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

netConnect("Stg4").Item(myArray(0)) = myArray(myArray.Length - 1)
msDef("Stg4")(myArray(0)) = New Dictionary(Of String, Double)

For Each combo As String In myArray
   Select Case combo.Trim(".").Split(".").Length
      Case 1
         msDef("Stg4")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
      Case Else
         Dim assetsInComb As String() = combo.Trim(".").Split(".")
         Dim maxProb As Double = 0

         For Each asst As String In assetsInComb
            If msDef("Stg4")(myArray(0)).Item("." & asst & ".") > maxProb Then
               maxProb = msDef("Stg4")(myArray(0)).Item("." & asst & ".")
            End If
         Next

         msDef("Stg4")(myArray(0)).Item(combo) = maxProb
   End Select
Next
```

```
    msDef("Stg4")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage5"
nextLine = sr.ReadLine() 'read in the first asset in "Stage5"

While stageNotDone
    If nextLine.IndexOf(",,") <> -1 Then
        nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
    Else
        nextLine = nextLine.Trim(",")
    End If

    'Retrieves the assets connected to the current asset
    Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                              nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

    netConnect("Stg5").Item(myArray(0)) = myArray(myArray.Length - 1)
    msDef("Stg5")(myArray(0)) = New Dictionary(Of String, Double)

    For Each combo As String In myArray
        Select Case combo.Trim(".").Split(".").Length
```

```
            Case 1
               msDef("Stg5")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
            Case Else
               Dim assetsInComb As String() = combo.Trim(".").Split(".")
               Dim maxProb As Double = 0

               For Each asst As String In assetsInComb
                  If msDef("Stg5")(myArray(0)).Item("." & asst & ".") > maxProb Then
                     maxProb = msDef("Stg5")(myArray(0)).Item("." & asst & ".")
                  End If
               Next

               msDef("Stg5")(myArray(0)).Item(combo) = maxProb
         End Select
      Next

      msDef("Stg5")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

      nextLine = sr.ReadLine
      If nextLine.IndexOf("endStage") <> -1 Then
         stageNotDone = False
      End If
   End While

   stageNotDone = True
   sr.ReadLine() 'read in "Stage6"
   nextLine = sr.ReadLine() 'read in the first asset in "Stage6"

   While stageNotDone
```

```
If nextLine.IndexOf(",,") <> -1 Then
   nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
Else
   nextLine = nextLine.Trim(",")
End If

'Retrieves the assets connected to the current asset
Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                          nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

netConnect("Stg6").Item(myArray(0)) = myArray(myArray.Length - 1)
msDef("Stg6")(myArray(0)) = New Dictionary(Of String, Double)

For Each combo As String In myArray
   Select Case combo.Trim(".").Split(".").Length
      Case 1
         msDef("Stg6")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
      Case Else
         Dim assetsInComb As String() = combo.Trim(".").Split(".")
         Dim maxProb As Double = 0

         For Each asst As String In assetsInComb
            If msDef("Stg6")(myArray(0)).Item("." & asst & ".") > maxProb Then
               maxProb = msDef("Stg6")(myArray(0)).Item("." & asst & ".")
            End If
         Next

         msDef("Stg6")(myArray(0)).Item(combo) = maxProb
   End Select
```

```
    Next

    msDef("Stg6")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage7"
nextLine = sr.ReadLine() 'read in the first asset in "Stage7"

While stageNotDone
    If nextLine.IndexOf(",,") <> -1 Then
        nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
    Else
        nextLine = nextLine.Trim(",")
    End If

    'Retrieves the assets connected to the current asset
    Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                              nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

    netConnect("Stg7").Item(myArray(0)) = myArray(myArray.Length - 1)
    msDef("Stg7")(myArray(0)) = New Dictionary(Of String, Double)

    For Each combo As String In myArray
```

```
        Select Case combo.Trim(".").Split(".").Length
            Case 1
                msDef("Stg7")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
            Case Else
                Dim assetsInComb As String() = combo.Trim(".").Split(".")
                Dim maxProb As Double = 0

                For Each asst As String In assetsInComb
                    If msDef("Stg7")(myArray(0)).Item("." & asst & ".") > maxProb Then
                        maxProb = msDef("Stg7")(myArray(0)).Item("." & asst & ".")
                    End If
                Next

                msDef("Stg7")(myArray(0)).Item(combo) = maxProb
        End Select
    Next

    msDef("Stg7")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage8"
nextLine = sr.ReadLine() 'read in the first asset in "Stage8"
```

```
While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If

   'Retrieves the assets connected to the current asset
   Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                        nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

   netConnect("Stg8").Item(myArray(0)) = myArray(myArray.Length - 1)
   msDef("Stg8")(myArray(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray
      Select Case combo.Trim(".").Split(".").Length
         Case 1
            msDef("Stg8")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
         Case Else
            Dim assetsInComb As String() = combo.Trim(".").Split(".")
            Dim maxProb As Double = 0

            For Each asst As String In assetsInComb
               If msDef("Stg8")(myArray(0)).Item("." & asst & ".") > maxProb Then
                  maxProb = msDef("Stg8")(myArray(0)).Item("." & asst & ".")
               End If
            Next

            msDef("Stg8")(myArray(0)).Item(combo) = maxProb
```

```
        End Select
    Next

    msDef("Stg8")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage9"
nextLine = sr.ReadLine() 'read in the first asset in "Stage9"

While stageNotDone
    If nextLine.IndexOf(",,") <> -1 Then
        nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
    Else
        nextLine = nextLine.Trim(",")
    End If

    'Retrieves the assets connected to the current asset
    Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                              nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

    netConnect("Stg9").Item(myArray(0)) = myArray(myArray.Length - 1)
    msDef("Stg9")(myArray(0)) = New Dictionary(Of String, Double)
```

```
    For Each combo As String In myArray
        Select Case combo.Trim(".").Split(".").Length
            Case 1
                msDef("Stg9")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
            Case Else
                Dim assetsInComb As String() = combo.Trim(".").Split(".")
                Dim maxProb As Double = 0

                For Each asst As String In assetsInComb
                    If msDef("Stg9")(myArray(0)).Item("." & asst & ".") > maxProb Then
                        maxProb = msDef("Stg9")(myArray(0)).Item("." & asst & ".")
                    End If
                Next

                msDef("Stg9")(myArray(0)).Item(combo) = maxProb
        End Select
    Next

    msDef("Stg9")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage10"
nextLine = sr.ReadLine() 'read in the first asset in "Stage10"
```

```
While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If

   'Retrieves the assets connected to the current asset
   Dim myArray As String() = GenSystem.GenCombinations(nextLine.Substring(0, nextLine.IndexOf(",")), _
                             nextLine.Substring(nextLine.IndexOf(",") + 1).Split(","))

   netConnect("Stg10").Item(myArray(0)) = myArray(myArray.Length - 1)
   msDef("Stg10")(myArray(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray
      Select Case combo.Trim(".").Split(".").Length
         Case 1
            msDef("Stg10")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
         Case Else
            Dim assetsInComb As String() = combo.Trim(".").Split(".")
            Dim maxProb As Double = 0

            For Each asst As String In assetsInComb
               If msDef("Stg10")(myArray(0)).Item("." & asst & ".") > maxProb Then
                  maxProb = msDef("Stg10")(myArray(0)).Item("." & asst & ".")
               End If
            Next
```

```
            msDef("Stg10")(myArray(0)).Item(combo) = maxProb
        End Select
    Next

    msDef("Stg10")(myArray(0)).Item(myArray(0)) = 0 'this value is calculated

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While

stageNotDone = True
sr.ReadLine() 'read in empty line
sr.ReadLine() 'read in 'TASK2"
sr.ReadLine() 'read in "Stage2"
nextLine = sr.ReadLine() 'read in the first asset in "Stage2"

While stageNotDone
    If nextLine.IndexOf(",,") <> -1 Then
        nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
    Else
        nextLine = nextLine.Trim(",")
    End If

    Dim myArray As String() = GenSystem.GenCombinations(nextLine, New String() {})

    spCol("Stg2")(myArray(0)) = New Dictionary(Of String, Double)
```

```
   For Each combo As String In myArray
      spCol("Stg2")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
   Next

   Dim myArray2 As String() = GenSystem.GenCombinations(nextLine, New String() {nextLine})

   spCol("Stg4")(myArray2(0)) = New Dictionary(Of String, Double)
   spCol("Stg5")(myArray2(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray2
      If myArray2(0) = combo Then
         spCol("Stg4")(myArray2(0)).Item(combo) = 0
         spCol("Stg5")(myArray2(0)).Item(combo) = 0
      Else
         spCol("Stg4")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
         spCol("Stg5")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
      End If
   Next

   nextLine = sr.ReadLine
   If nextLine.IndexOf("endStage") <> -1 Then
      stageNotDone = False
   End If
End While

'Skip over all the extra Task2 stage connections I don't need.
While nextLine.IndexOf("TASK3") = -1
   nextLine = sr.ReadLine()
End While
```

```
stageNotDone = True
sr.ReadLine() 'read in "Stage2"
nextLine = sr.ReadLine() 'read in the first asset in "Stage2"

While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If

   Dim myArray As String() = GenSystem.GenCombinations(nextLine, New String() {})

   intCol("Stg2")(myArray(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray
      intCol("Stg2")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
   Next

   Dim myArray2 As String() = GenSystem.GenCombinations(nextLine, New String() {nextLine})

   intCol("Stg4")(myArray2(0)) = New Dictionary(Of String, Double)
   intCol("Stg5")(myArray2(0)) = New Dictionary(Of String, Double)

   For Each combo As String In myArray2
      If myArray2(0) = combo Then
         intCol("Stg4")(myArray2(0)).Item(combo) = 0
         intCol("Stg5")(myArray2(0)).Item(combo) = 0
```

```
      Else
         intCol("Stg4")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
         intCol("Stg5")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
      End If

   Next

   nextLine = sr.ReadLine
   If nextLine.IndexOf("endStage") <> -1 Then
      stageNotDone = False
   End If
End While

'Skip over all the extra Task3 stage connections I don't need.
While nextLine.IndexOf("TASK4") = -1
   nextLine = sr.ReadLine()
End While

stageNotDone = True
sr.ReadLine() 'read in "Stage2"
nextLine = sr.ReadLine() 'read in the first asset in "Stage2"

While stageNotDone
   If nextLine.IndexOf(",,") <> -1 Then
      nextLine = nextLine.Substring(0, nextLine.IndexOf(",,"))
   Else
      nextLine = nextLine.Trim(",")
   End If
```

```
    Dim myArray As String() = GenSystem.GenCombinations(nextLine, New String() {})

    topSec("Stg2")(myArray(0)) = New Dictionary(Of String, Double)

    For Each combo As String In myArray
        topSec("Stg2")(myArray(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
    Next

    Dim myArray2 As String() = GenSystem.GenCombinations(nextLine, New String() {nextLine})

    topSec("Stg4")(myArray2(0)) = New Dictionary(Of String, Double)
    topSec("Stg5")(myArray2(0)) = New Dictionary(Of String, Double)

    For Each combo As String In myArray2
        If myArray2(0) = combo Then
            topSec("Stg4")(myArray2(0)).Item(combo) = 0
            topSec("Stg5")(myArray2(0)).Item(combo) = 0
        Else
            topSec("Stg4")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
            topSec("Stg5")(myArray2(0)).Item(combo) = GenRandProb(randNum.NextDouble, lowerBnd, range)
        End If
    Next

    nextLine = sr.ReadLine
    If nextLine.IndexOf("endStage") <> -1 Then
        stageNotDone = False
    End If
End While
```

```
        sr.Close()
    End Using

    Console.WriteLine("Network information sucessfully imported from " & networkCSVFile & ".csv.")
    Console.WriteLine("Probabilities successfully generated." & vbCrLf)
End Sub


'Reads in the probabilities for a network that was specified using the ImportConnections sub.

Public Shared Sub ImportNetStructAndProbs(ByVal probCSVFile As String, _
                ByRef msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef netConnect As Dictionary(Of String, Dictionary(Of String, String)))

    '*NOTE* A *.csv file named that contains the network connections and all combinations and the associated probabilities
    'filled in must be present in the ...\GSMOpt\bin\Release project directory.
    'This file has a special format.


    'Dim theFile() As String = File.ReadAllLines("networkProbsEntered.csv")
    Dim theFile() As String = File.ReadAllLines(probCSVFile & ".csv")
    Dim idx As Integer = 1 'start with the second line of the file

    'Task 1, Stage 2
    idx += 1 'skip the "Stage2" line
    While theFile(idx).IndexOf("endStage") = -1
```

```
    Dim asset As String = theFile(idx).Split(",")(0)
    msDef("Stg2")(asset) = New Dictionary(Of String, Double)

    msDef("Stg2")(asset).Add(theFile(idx).Split(",")(0), CDbl(theFile(idx).Split(",")(1)))
    idx += 1 'skip to "endCombinations" line
    idx += 1 'skip to next asset
End While

idx += 1 'skip to "Stage3" line

'Task 1, all other stages
For Each stg As String In msDef.Keys
    If stg <> "Stg2" Then
        idx += 1 'skip the "Stagei" line

        While theFile(idx).IndexOf("endStage") = -1
            Dim asset As String = theFile(idx).Split(",")(0)
            msDef(stg)(asset) = New Dictionary(Of String, Double)

            While theFile(idx).IndexOf("endCombinations") = -1
                msDef(stg)(asset).Add(theFile(idx).Split(",")(0), CDbl(theFile(idx).Split(",")(1)))
                idx += 1
            End While

            netConnect(stg).Add(asset, theFile(idx - 1).Split(",")(0))
            idx += 1 'skip the "endCombinations" line
        End While
        idx += 1 'skip the "endStage" line
    End If
```

```
Next

idx += 2 'skip the white line and "TASK2" line

For Each stg As String In spCol.Keys
   idx += 1 'skip the "Stagei" line

   While theFile(idx).IndexOf("endStage") = -1
      Dim asset As String = theFile(idx).Split(",")(0)
      spCol(stg)(asset) = New Dictionary(Of String, Double)

      While theFile(idx).IndexOf("endCombinations") = -1
         spCol(stg)(asset).Add(theFile(idx).Split(",")(0), CDbl(theFile(idx).Split(",")(1)))
         idx += 1
      End While
      idx += 1 'skip the "endCombinations" line
   End While
   idx += 1 'skip the "endStage" line
Next

idx += 2 'skip the white line and "TASK3" line

For Each stg As String In intCol.Keys
   idx += 1 'skip the "Stagei" line

   While theFile(idx).IndexOf("endStage") = -1
      Dim asset As String = theFile(idx).Split(",")(0)
      intCol(stg)(asset) = New Dictionary(Of String, Double)
```

```
            While theFile(idx).IndexOf("endCombinations") = -1
                intCol(stg)(asset).Add(theFile(idx).Split(",")(0), CDbl(theFile(idx).Split(",")(1)))
                idx += 1
            End While
            idx += 1 'skip the "endCombinations" line
        End While
        idx += 1 'skip the "endStage" line
    Next

    idx += 2 'skip the white line and "TASK4" line

    For Each stg As String In topSec.Keys
        idx += 1 'skip the "Stagei" line

        While theFile(idx).IndexOf("endStage") = -1
            Dim asset As String = theFile(idx).Split(",")(0)
            topSec(stg)(asset) = New Dictionary(Of String, Double)

            While theFile(idx).IndexOf("endCombinations") = -1
                topSec(stg)(asset).Add(theFile(idx).Split(",")(0), CDbl(theFile(idx).Split(",")(1)))
                idx += 1
            End While
            idx += 1 'skip the "endCombinations" line
        End While
        idx += 1 'skip the "endStage" line
    Next

    Console.WriteLine("Network structure and probability information imported successfully." & vbCrLf)
End Sub
```

```
'This sub saves both the network connections and the probabilities of the network to a
'.csv file. This means that the whole network and it's associated probabilities do not
'have to be loaded each time.

 Public Shared Sub SaveProbabilities( _
                ByRef msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                ByRef netConnect As Dictionary(Of String, Dictionary(Of String, String)))

    Dim stgDict As Dictionary(Of String, String) = New Dictionary(Of String, String)
    stgDict.Add("Stg2", "Stage2")
    stgDict.Add("Stg3", "Stage3")
    stgDict.Add("Stg4", "Stage4")
    stgDict.Add("Stg5", "Stage5")
    stgDict.Add("Stg6", "Stage6")
    stgDict.Add("Stg7", "Stage7")
    stgDict.Add("Stg8", "Stage8")
    stgDict.Add("Stg9", "Stage9")
    stgDict.Add("Stg10", "Stage10")

    Console.Write("Name of file to store network probabilites in: ")
    Dim fileName2 As String = Console.ReadLine()

    Console.WriteLine("The system data will be saved in " & fileName2 & ".csv")
```

```
'Write the probability information to the specified file.
Using sw2 As StreamWriter = New StreamWriter(fileName2 & ".csv")
   sw2.WriteLine("TASK1")
   For Each stg As String In msDef.Keys
      sw2.WriteLine(stgDict(stg)) 'write the name of the stage
      For Each asset As String In msDef(stg).Keys
         For Each combo As String In msDef(stg)(asset).Keys
            sw2.WriteLine(combo & "," & msDef(stg)(asset).Item(combo))
         Next
         sw2.WriteLine("endCombinations") 'add a white line between each asset and its combinations
      Next
      sw2.WriteLine("endStage")
   Next

   sw2.WriteLine()
   sw2.WriteLine("TASK2")
   For Each stg As String In spCol.Keys
      sw2.WriteLine(stgDict(stg)) 'write the name of the stage
      For Each asset As String In spCol(stg).Keys
         For Each combo As String In spCol(stg)(asset).Keys
            sw2.WriteLine(combo & "," & spCol(stg)(asset).Item(combo))
         Next
         sw2.WriteLine("endCombinations") 'add a white line between each asset and its combinations
      Next
      sw2.WriteLine("endStage")
   Next

   sw2.WriteLine()
   sw2.WriteLine("TASK3")
```

```
        For Each stg As String In intCol.Keys
            sw2.WriteLine(stgDict(stg)) 'write the name of the stage
            For Each asset As String In intCol(stg).Keys
                For Each combo As String In intCol(stg)(asset).Keys
                    sw2.WriteLine(combo & "," & intCol(stg)(asset).Item(combo))
                Next
                sw2.WriteLine("endCombinations") 'add a white line between each asset and its combinations
            Next
            sw2.WriteLine("endStage")
        Next

        sw2.WriteLine()
        sw2.WriteLine("TASK4")
        For Each stg As String In topSec.Keys
            sw2.WriteLine(stgDict(stg)) 'write the name of the stage
            For Each asset As String In topSec(stg).Keys
                For Each combo As String In topSec(stg)(asset).Keys
                    sw2.WriteLine(combo & "," & topSec(stg)(asset).Item(combo))
                Next
                sw2.WriteLine("endCombinations") 'add a white line between each asset and its combinations
            Next
            sw2.WriteLine("endStage")
        Next

        sw2.Close()
    End Using
End Sub
```

```
'This sub randomly generates the probabilities in the specified range for every asset in the
'network.

 Public Shared Sub RegenProbs(ByVal lowerBnd2 As Double, ByVal upperBnd2 As Double, _
                              ByVal lowerBnd As Double, ByVal upperBnd As Double, _
                              ByRef msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                              ByRef spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                              ByRef intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                              ByRef topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

    Dim range As Double = upperBnd - lowerBnd
    Dim randNum As New Random()

    'Task 1, Stage 2
    For Each asset As String In msDef("Stg2").Keys
        Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

        For Each comb As String In msDef("Stg2")(asset).Keys
            tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd2, (upperBnd2 - lowerBnd2)))
        Next

        For Each comb As String In tempDict.Keys
            msDef("Stg2")(asset).Item(comb) = tempDict.Item(comb)
        Next
    Next

    'Task 1, Stages 3-10
    For Each stg As String In msDef.Keys
        If stg <> "Stg2" Then
```

```
For Each asset As String In msDef(stg).Keys
   Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

   For Each comb As String In msDef(stg)(asset).Keys
      If comb = asset Then
         tempDict.Add(comb, 0)
      Else
         Select Case comb.Trim(".").Split(".").Length
            Case 1
               tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
            Case Else
               Dim assetsInComb As String() = comb.Trim(".").Split(".")
               Dim maxProb As Double = 0

               For Each asst As String In assetsInComb
                  If tempDict.Item("." & asst & ".") > maxProb Then
                     maxProb = tempDict.Item("." & asst & ".")
                  End If
               Next

               tempDict.Add(comb, maxProb)
         End Select
      End If
   Next

   For Each comb As String In tempDict.Keys
      msDef(stg)(asset).Item(comb) = tempDict.Item(comb)
   Next
Next
```

```
    End If
Next

'Task 2
For Each asset As String In spCol("Stg2").Keys
    Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

    For Each comb As String In spCol("Stg2")(asset).Keys
        tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
    Next

    For Each comb As String In tempDict.Keys
        spCol("Stg2")(asset).Item(comb) = tempDict.Item(comb)
    Next
Next

For Each stg As String In spCol.Keys
    If stg <> "Stg2" Then
        For Each asset As String In spCol(stg).Keys
            Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

            For Each comb As String In spCol(stg)(asset).Keys
                If comb = asset Then
                    tempDict.Add(comb, 0)
                Else
                    tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
                End If
            Next
```

```
            For Each comb As String In tempDict.Keys
                spCol(stg)(asset).Item(comb) = tempDict.Item(comb)
            Next
        Next
    End If
Next

'Task 3
For Each asset As String In intCol("Stg2").Keys
    Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

    For Each comb As String In intCol("Stg2")(asset).Keys
        tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
    Next

    For Each comb As String In tempDict.Keys
        intCol("Stg2")(asset).Item(comb) = tempDict.Item(comb)
    Next
Next

For Each stg As String In intCol.Keys
    If stg <> "Stg2" Then
        For Each asset As String In intCol(stg).Keys
            Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

            For Each comb As String In intCol(stg)(asset).Keys
                If comb = asset Then
                    tempDict.Add(comb, 0)
                Else
```

```
                tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
            End If
        Next

        For Each comb As String In tempDict.Keys
            intCol(stg)(asset).Item(comb) = tempDict.Item(comb)
        Next
    Next
  End If
Next

'Task 4
For Each asset As String In topSec("Stg2").Keys
  Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

  For Each comb As String In topSec("Stg2")(asset).Keys
    tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
  Next

  For Each comb As String In tempDict.Keys
    topSec("Stg2")(asset).Item(comb) = tempDict.Item(comb)
  Next
Next

For Each stg As String In topSec.Keys
  If stg <> "Stg2" Then
    For Each asset As String In topSec(stg).Keys
      Dim tempDict As Dictionary(Of String, Double) = New Dictionary(Of String, Double)
```

```
                For Each comb As String In topSec(stg)(asset).Keys
                    If comb = asset Then
                        tempDict.Add(comb, 0)
                    Else
                        tempDict.Add(comb, GenRandProb(randNum.NextDouble, lowerBnd, range))
                    End If
                Next

                For Each comb As String In tempDict.Keys
                    topSec(stg)(asset).Item(comb) = tempDict.Item(comb)
                Next
            Next
        End If
    Next
End Sub


'This function generates a random number less than 1 which is in the specified range.

Private Shared Function GenRandProb(ByVal randNumber As Double, ByVal lowerBnd As Double, _
                                    ByVal range As Double) As Double

    Return (lowerBnd + (randNumber * range))
End Function

End Class
```

B.3. GenSystem.vb

```
Public Class GenSystem
  'Function GenCombinations takes an array of assets and generates all possible combinations of that
  'array of assets. Input 'asset' is placed at the front of the array that is returned, and all
  'combinations are placed in subsequent cells. The combinations are sorted from least number of
  'elements in the combination (1) to most (myArray.Length).

  Public Shared Function GenCombinations(ByVal asset As String, ByVal connectedAssets As String()) As String()
    'The maximum number of elements in any combination generated is the total
    'number of elements in group.
    Dim a As String() = New String(connectedAssets.Length - 1) {}

    'The sum of each row n in Pascals triangle is equal to 2^n
    'theCombs contains combinations of 2 elements or more.
    Dim theCombs As String() = New String((2 ^ connectedAssets.Length) - 2 - connectedAssets.Length) {}
    Dim numCommas As Integer() = New Integer(theCombs.Length - 1) {}

    Dim idx As Integer = 0

    'Pseudocode for this algorithm is taken from http://tovganesh.googlepages.com/comb.pdf
    ' and generates all combinations of 2 or more.
    Dim n As Integer = connectedAssets.Length - 1

    For i As Integer = 0 To n
      For j As Integer = i + 1 To n
        a(0) = i
        a(1) = j
        theCombs(idx) = "." & connectedAssets(a(0)) & "." & connectedAssets(a(1)) & "."
```

```
numCommas(idx) = 1

If n >= 2 Then
  Dim t As Integer = 1
  Dim pos As Integer = 1
  Dim m As Integer = pos + 1
  Dim done = 1

  While done
    For k As Integer = a(pos) + 1 To n
      idx += 1
      a(m) = k

      theCombs(idx) = "."
      For z As Integer = 0 To m - 1
        theCombs(idx) &= connectedAssets(a(z)) & "."
        numCommas(idx) += 1
      Next
      theCombs(idx) &= connectedAssets(a(m)) & "."

      m += 1
    Next

    pos = m - 1
    m = pos

    If (a(pos) = n - 1) Then
      t += 1
    Else
```

```
            t = 1
          End If

          If pos = 1 Then
            done = 0
          End If
        End While

        idx += 1
      End If
    Next
  Next

  Return SortCombinations(asset, connectedAssets, theCombs, numCommas)
End Function


'Function SortCombinations takes an asset and and array containing it's list combination of 2
'or more elements. It places the input'asset' at the front of the returned array, followed by all
'1 element combinations, followed by all combinations of 2 or more elements. It also sorts the
'combinations in order of least number of elements (1) at the start of the array, to most number of
'elements (connectedAssets.Length) at the end of the array.

 Private Shared Function SortCombinations(ByVal asset As String, ByVal connectedAssets As String(), _
                                          ByVal thecombs As String(), ByVal numCommas As Integer()) As String()

  Dim sortedCombs As String() = New String((2 ^ connectedAssets.Length) - 1) {}

  'Place the asset id in the first element.
```

```
  sortedCombs(0) = asset

  'Place all the combinations of 1 element first.
  For i As Integer = 0 To connectedAssets.Length - 1
    sortedCombs(i + 1) = "." & connectedAssets(i) & "."
  Next

  Dim idx As Integer = connectedAssets.Length + 1

  'Sort the combinations by number of elements in the combination.
  For i As Integer = 0 To connectedAssets.Length - 1
    For j As Integer = 0 To thecombs.Length - 1
      If numCommas(j) = i Then
        sortedCombs(idx) = thecombs(j)
        idx += 1
      End If
    Next
  Next
  Return sortedCombs
End Function


'Function FindAssignable searches the input network to determine which assets are capable of
'working on different tasks by by checking to see if an asset that appears in stage two of task 1
' also appears in stage two of any other task type.

 Public Shared Function FindAssignable( _
              ByVal msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
              ByVal spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
```

```
            ByVal intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

Dim assignables As String() = New String(0) {}
Dim idx As Integer = 0
Dim found As Boolean = False

'Take each asset in stage 2 of task 1 and check if it appears in stage 2 of any other task.
For Each asst As String In msDef("Stg2").Keys
  'Check for the asset in task 2 (Space Collect).
  For Each asstTsk2 As String In spCol("Stg2").Keys
    If String.Equals(asst, asstTsk2) Then
      assignables(idx) = asst
      idx += 1
      found = True
      Exit For
    End If
  Next

  'If the asset was not found in task 2, check task 3 (Intelligence Collect).
  If Not found Then
    For Each asstTsk3 As String In intCol("Stg2").Keys
      If String.Equals(asst, asstTsk3) Then
        assignables(idx) = asst
        idx += 1
        found = True
        Exit For
      End If
    Next
```

```
        End If

        'If the asset was not found in task 3, check task 4 (Classified).
        If Not found Then
          For Each asstTsk4 As String In topSec("Stg2").Keys
            If String.Equals(asst, asstTsk4) Then
              assignables(idx) = asst
              idx += 1
              found = True
              Exit For
            End If
          Next
        End If

        'if a new asset was added, redimensions the array.
        If found = True Then
          ReDim Preserve assignables(assignables.Length)
        End If

        found = False 'set found to false for the next asset check
      Next

      ReDim Preserve assignables(assignables.Length - 2)
      Return assignables
    End Function

End Class
```

B.4. GenNetwork.vb

```
Public Class GenNetwork
  'NewNetwork takes the data arrays the represent the general system, and a specific allocation of
  'assets to task 1, and creates new data arrays that represent the network under the specific
  'allocation. It then uses the modified arrays to calculate the necessary probabilities for that
  'network, and stores them in the specified file.

   Public Shared Function NewNetwork(ByVal networkAllocation As String(), ByVal assignable As String(), _
            ByVal msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal netConnect As Dictionary(Of String, Dictionary(Of String, String))) As Double()

      Dim msDefAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
          New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim spColAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
          New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim intColAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
          New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim topSecAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))) = _
          New Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double)))

      Dim netConnectAlloc As Dictionary(Of String, Dictionary(Of String, String)) = _
          New Dictionary(Of String, Dictionary(Of String, String))
```

```
InitDictionaries(msDefAlloc, spColAlloc, intColAlloc, topSecAlloc)

CopyDictionaries(msDef, spCol, intCol, topSec, msDefAlloc, spColAlloc, intColAlloc, topSecAlloc)

'Check to see if no assignable assets were assigned to task 1, create modified task 1
'arrays and copy the task 2, 3, and 4 arrays.
If UBound(networkAllocation) = -1 Then
   'The other task dictionaries stay the same, only task 1 is changed.
   ModTask1(assignable, msDefAlloc) 'All assignable assets are not on task 1
   ModNetConnect(assignable, netConnect, netConnectAlloc)

Else 'otherwise at least one assignable asset was assigned to task 1
   'If all assets were assigned to task 1, only modify the task 2, 3 and 4 arrays
   If networkAllocation.Length = assignable.Length Then
      'The task 1 dictionary stays the same, but we need to populate the
      'netConnectAlloc dictionary.
      ModNetConnect(New String() {}, netConnect, netConnectAlloc)

      'The other task dictionaries need to be updated.
      ModOtherTasks(networkAllocation, spColAlloc)
      ModOtherTasks(networkAllocation, intColAlloc)
      ModOtherTasks(networkAllocation, topSecAlloc)

   Else 'Need to modify all tasks.
      Dim notTask1 As String() = New String(assignable.Length - networkAllocation.Length - 1) {}
      Dim idx As Integer = 0
      Dim notPart As Boolean = True 'assume the asset is not part of task 1
```

```
        'Place all assets not assigned to task 1 in this array.
        For i As Integer = 0 To assignable.Length - 1
            'Checks the current asset against all assets that are not assigned
            'to this task. If the current asset matches an asset not assigned to
            'this task, then set isPart to false, so that the asset is not placed
            'in the modified array.
            For j As Integer = 0 To networkAllocation.Length - 1
                If String.Equals(networkAllocation(j), assignable(i)) Then
                    notPart = False
                    Exit For
                End If
            Next

            If notPart Then
                notTask1(idx) = assignable(i)
                idx += 1
            End If

            notPart = True
        Next

        ModTask1(notTask1, msDefAlloc)
        ModNetConnect(notTask1, netConnect, netConnectAlloc)

        ModOtherTasks(networkAllocation, spColAlloc)
        ModOtherTasks(networkAllocation, intColAlloc)
        ModOtherTasks(networkAllocation, topSecAlloc)
    End If
End If
```

```
    'Calculate the probabilities for the specific network allocation.
    CalcProbabilities.Task1Probs(msDefAlloc, netConnectAlloc)
    CalcProbabilities.OtherTaskProbs(spColAlloc)
    CalcProbabilities.OtherTaskProbs(intColAlloc)
    CalcProbabilities.OtherTaskProbs(topSecAlloc)

    'Print the probability of success for each asset in each stage of task 1 to a .csv file.
    If UBound(networkAllocation) = -1 Then
        ExportTask1AssetProbs(True, "No assets assigned to task 1.", msDefAlloc, msDef)
    Else
        Dim allocation As String = "."

        For Each asset As String In networkAllocation
            allocation &= asset & "."
        Next

        ExportTask1AssetProbs(False, allocation, msDefAlloc, msDef)
    End If

    Return EndStageProbs(msDefAlloc("Stg10").Count + 3 - 1, msDefAlloc("Stg10"), _
                spColAlloc("Stg5"), intColAlloc("Stg5"), topSecAlloc("Stg5"))
End Function


'ModTask1 takes the assets not assigned to task 1 in this network allocation and
'removes them and all combinations they belong to from the network.

Private Shared Sub ModTask1(ByVal notTask1 As String(), _
```

```
    ByRef msDefAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

For Each stg As String In msDefAlloc.Keys
   Select Case stg
      Case "Stg3", "Stg4", "Stg5", "Stg6"
         'First remove any assets that are not assigned to task 1.
         For k As Integer = 0 To notTask1.Length - 1
            'If the k'th asset not allocated to task 1 is present in the current stage of
            'task 1, remove it.
            If msDefAlloc(stg).ContainsKey(notTask1(k)) Then
               msDefAlloc(stg).Remove(notTask1(k))
            End If
         Next

         'For each remaining asset in the stage, search the assets combinations and
         'remove any combination that has an asset not assigned to task 1.
         For Each asset As String In msDefAlloc(stg).Keys
            Dim combToDel As Dictionary(Of String, Double) = New Dictionary(Of String, Double)

            For j As Integer = 0 To notTask1.Length - 1
               For Each comb As String In msDefAlloc(stg)(asset).Keys
                  If comb.IndexOf("." & notTask1(j) & ".") <> -1 Then
                     'Combination does contain the asset, so place it on the list
                     'of combinations to remove.
                     combToDel.Item(comb) = 0
                  End If
               Next
            Next
```

```
                For Each comb As String In combToDel.Keys
                    msDefAlloc(stg)(asset).Remove(comb)
                Next
            Next

            ' " "Case "Stg7", "Stg8", "Stg9", "Stg10"
            ' " "From the network assumptions, no assignable asset is connected to any asset
            ' " "in these stages, so they are the same as in the general system.

        Case "Stg2"
            'If the k'th asset not allocated to task 1 is present in stage 2 of
            'task 1, remove it.
            For k As Integer = 0 To notTask1.Length - 1
                If msDefAlloc(stg).ContainsKey(notTask1(k)) Then
                    msDefAlloc(stg).Remove(notTask1(k))
                End If
            Next
    End Select
Next

End Sub


'ModNetConnect modifies the netConnectAlloc dictionary to reflect the current
'network allocation of assignable assets.

Private Shared Sub ModNetConnect(ByVal notTask1 As String(), _
    ByVal netConnect As Dictionary(Of String, Dictionary(Of String, String)), _
    ByRef netConnectAlloc As Dictionary(Of String, Dictionary(Of String, String)))
```

```
For Each stg As String In netConnect.Keys
  netConnectAlloc(stg) = New Dictionary(Of String, String)(netConnect(stg))
Next

'Modify netConnectAlloc only if any assignable asset has been assigned to another task.
If UBound(notTask1) <> -1 Then
  Dim stageToMod As String() = New String() {"Stg3", "Stg4", "Stg5", "Stg6"}

  For Each stg As String In netConnectAlloc.Keys
    'First remove any assets that are not assigned to task 1.
    For k As Integer = 0 To notTask1.Length - 1
      'If the k'th asset not allocated to task 1 is present in the current stage of
      'task 1, remove it.
      If netConnectAlloc(stg).ContainsKey(notTask1(k)) Then
        netConnectAlloc(stg).Remove(notTask1(k))
      End If
    Next

    Dim modconnect As New Dictionary(Of String, String)
    modconnect = New Dictionary(Of String, String)(netConnectAlloc(stg))

    For Each asset As String In netConnectAlloc(stg).Keys
      For Each mch As String In notTask1
        If modconnect(asset).IndexOf("." & mch & ".") <> -1 Then
          '*NOTE* This piece of code will produce a combination of only a
          'single comma if the current asset is only connected to an assignable
          'asset that is assigned to another task
          modconnect.Item(asset) = modconnect(asset).Replace("." & mch & ".", ".")
```

```
            End If
          Next
        Next

        For Each asset As String In modconnect.Keys
          netConnectAlloc(stg).Item(asset) = modconnect(asset)
        Next
      Next
    End If

End Sub


'ModOtherTasks creates modified network arrays for tasks 2, 3 and 4. Due to the linear structure of
'these networks, if an asset work in stage 2 of any of these task, then it works in all other
'stages, and if it does not work in stage 2, then it works in no other stages.

Private Shared Sub ModOtherTasks(ByVal onTask1 As String(), _
        ByRef tskAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

  'Check each asset that is assigned to task 1 against each asset that is
  'currently working on task 2, 3, or 4. If the asset that is working on
  'task 1 is currently working on task 2, 3, or 4, it is removed from this task.
  For j As Integer = 0 To onTask1.Length - 1
    For Each asst As String In tskAlloc("Stg2").Keys
      If String.Equals(onTask1(j), asst) Then
        tskAlloc("Stg2").Remove(asst)
        tskAlloc("Stg4").Remove(asst)
        tskAlloc("Stg5").Remove(asst)
```

```
            Exit For
          End If
        Next
      Next
    End Sub


    'CopyDictionaries places all the network information into the allocation dictionaries
    'which can then be modified to represent a specific network allocation.

     Private Shared Sub CopyDictionaries( _
                    ByVal msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByVal spCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByVal intCol As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByVal topSec As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByRef msDefAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByRef spColAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByRef intColAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
                    ByRef topSecAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

        For Each stg As String In msDef.Keys
            For Each asset As String In msDef(stg).Keys
                msDefAlloc(stg)(asset) = New Dictionary(Of String, Double)(msDef(stg)(asset))
            Next
        Next

        For Each stg As String In spCol.Keys
            For Each asset As String In spCol(stg).Keys
                spColAlloc(stg)(asset) = New Dictionary(Of String, Double)(spCol(stg)(asset))
```

```
        Next
    Next

    For Each stg As String In intCol.Keys
        For Each asset As String In intCol(stg).Keys
            intColAlloc(stg)(asset) = New Dictionary(Of String, Double)(intCol(stg)(asset))
        Next
    Next

    For Each stg As String In topSec.Keys
        For Each asset As String In topSec(stg).Keys
            topSecAlloc(stg)(asset) = New Dictionary(Of String, Double)(topSec(stg)(asset))
        Next
    Next

End Sub


'EndStageProbs places the end stage probabilities for a particular network allocation into an array.
'The individual probabilities of success for the Task 1 stage 10 assets are first placed in the array,
'followed by the overall probabilities of success for tasks 2-4.

Private Shared Function EndStageProbs(ByVal dimen As Integer, _
                                      ByVal task1 As Dictionary(Of String, Dictionary(Of String, Double)), _
                                      ByVal task2 As Dictionary(Of String, Dictionary(Of String, Double)), _
                                      ByVal task3 As Dictionary(Of String, Dictionary(Of String, Double)), _
                                      ByVal task4 As Dictionary(Of String, Dictionary(Of String, Double))) As Double()

    Dim probArray As Double() = New Double(dimen) {}
```

```
Dim index As Integer = 0

For Each asset As String In task1.Keys
  probArray(index) = task1(asset).Item(asset)
  index += 1
Next

'Task 2 overall probability of success.
Dim POF As Double = 1

For Each asset As String In task2.Keys
  POF *= (1 - task2(asset).Item(asset))
Next
probArray(index) = (1 - POF)
index += 1

'Task 3 overall probability of success.
POF = 1

For Each asset As String In task3.Keys
  POF *= (1 - task3(asset).Item(asset))
Next
probArray(index) = (1 - POF)
index += 1

'Task 4 overall probability of success.
POF = 1

For Each asset As String In task4.Keys
```

```
    POF *= (1 - task4(asset).Item(asset))
  Next
  probArray(index) = (1 - POF)

  Return probArray
End Function


'Description.

Private Shared Sub ExportTask1AssetProbs(ByVal noneAssigned As Boolean, ByVal networkAllocation As String, _
            ByVal task1 As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
            ByVal msDef As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

  Dim stageNames As String() = New String() {"Stg3", "Stg4", "Stg5", "Stg6", "Stg7", "Stg8", "Stg9", "Stg10"}

  If noneAssigned Then
    'Create the file "analysis.csv" and write the probabilities for when no
    'assignable assets are allocated to task 1.
    For Each stg As String In stageNames
      Using sw As StreamWriter = New StreamWriter(stg & ".csv")
        sw.Write("Assets assigned to task 1,")
        For Each asset As String In msDef(stg).Keys
          sw.Write(asset & ",")
        Next
        sw.WriteLine()
        sw.Write(networkAllocation & ",")
        For Each asset As String In msDef(stg).Keys
          Try
```

```
                sw.Write(task1(stg)(asset).Item(asset) & ",")
              Catch
                sw.Write(0 & ",")
              End Try
            Next
            sw.WriteLine()
          End Using
        Next

      Else
        For Each stg As String In stageNames
          Using sw As StreamWriter = New StreamWriter(stg & ".csv", True)
            sw.Write(networkAllocation & ",")
            For Each asset As String In msDef(stg).Keys
              Try
                sw.Write(task1(stg)(asset).Item(asset) & ",")
              Catch
                sw.Write(0 & ",")
              End Try
            Next
            sw.WriteLine()
          End Using
        Next
      End If
    End Sub

End Class
```

B.5. CalcProbabilities.vb

```
Public Class CalcProbabilities
  'Task1Probs calculates the probability of success for every asset in stages 3 - 10 of
  'task 1.

   Public Shared Sub Task1Probs( _
        ByRef msDefAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
        ByVal netConnectAlloc As Dictionary(Of String, Dictionary(Of String, String)))

      '*NOTE* Probabilities have to be calculated in sequential order, or else they will be wrong.

      'Calculate the probability of success for each asset in stages 4, 5, 6, 7, 8, 9, 10.
      Dim stg As String() = New String() {"Stg2", "Stg3", "Stg4", "Stg5", "Stg6", "Stg7", "Stg8", "Stg9", "Stg10"}

      For i As Integer = 1 To stg.Length - 1
         StageCalcs(msDefAlloc, netConnectAlloc, stg(i), stg(i - 1))
      Next

   End Sub


  'OtherTaskProbs calculates the probability of success for each asset in stages 4 and 5 of tasks
  '2, 3, and 4.

  Public Shared Sub OtherTaskProbs( _
    ByRef tskAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))))

    'Calculate the probability of success for each asset in stage 4.
```

```
  For Each asset As String In tskAlloc("Stg4").Keys
    tskAlloc("Stg4")(asset).Item(asset) = tskAlloc("Stg2")(asset)(asset) * tskAlloc("Stg4")(asset)("." & asset & ".")
  Next

  'Calculate the probability of success for each asset in stage 5.
  For Each asset As String In tskAlloc("Stg5").Keys
    tskAlloc("Stg5")(asset).Item(asset) = tskAlloc("Stg4")(asset)(asset) * tskAlloc("Stg5")(asset)("." & asset & ".")
  Next

End Sub


'StageCalcs calculates and stores the probability of success for every asset in the current
'stage. This sub performs the stage calculations for each stage in Task 1.

Private Shared Sub StageCalcs( _
    ByRef tskAlloc As Dictionary(Of String, Dictionary(Of String, Dictionary(Of String, Double))), _
    ByVal netConnectAlloc As Dictionary(Of String, Dictionary(Of String, String)), _
    ByVal currentStg As String, ByVal prevStg As String)

  For Each asset As String In tskAlloc(currentStg).Keys
    Dim pOS As Double = 0 'probability of success of current asset

    For Each comb As String In tskAlloc(currentStg)(asset).Keys
      If comb <> asset Then
        'pA is the probability that all the assets in the combination succeed, while all those
        'not in the combination but connected to the current asset fail.
        Dim pA As Double = 1
```

```
            'netConnect contains all assets connected to current asset.
            For Each mch As String In Split(netConnectAlloc(currentStg)(asset).Trim("."), ".")
              If comb.IndexOf("." & mch & ".") <> -1 Then
                'If the asset is in the combination  it should succeed.
                pA *= tskAlloc(prevStg)(mch)(mch)
              Else
                'If the asset is not in the combination it should fail.
                pA *= (1 - tskAlloc(prevStg)(mch)(mch))
              End If
            Next

            pOS += (tskAlloc(currentStg)(asset)(comb) * pA)
          End If
        Next

        tskAlloc(currentStg)(asset).Item(asset) = pOS
      Next
    End Sub

End Class
```

C. Data for Degradation Study

Table C.1, below, shows the probabilities assigned to the network used in section 3.4.

Table C.1: Probabilities Used for the Network in the Degradation Study

TASK1		Stage3 (cont.)		Stage3 (cont.)	
Stage2		.3.5.	0.798698	.1.5.10.	0.798698
1	0.965171	.3.6.	0.790476	.1.5.11.	0.798698
2	0.986536	.3.9.	0.755914	.1.6.9.	0.790476
3	0.945179	.3.10.	0.775777	.1.6.10.	0.790476
4	0.771135	.3.11.	0.75334	.1.6.11.	0.790476
5	0.796605	.4.5.	0.798698	.1.9.10.	0.775777
6	0.792256	.4.6.	0.790476	.1.9.11.	0.755914
7	0.757127	.4.9.	0.755914	.1.10.11.	0.775777
8	0.773516	.4.10.	0.775777	.2.3.4.	0.751817
9	0.965296	.4.11.	0.75334	.2.3.5.	0.798698
10	0.942627	.5.6.	0.798698	.2.3.6.	0.790476
11	0.965211	.5.9.	0.798698	.2.3.9.	0.755914
		.5.10.	0.798698	.2.3.10.	0.775777
Stage3		.5.11.	0.798698	.2.3.11.	0.75334
17		.6.9.	0.790476	.2.4.5.	0.798698
.1.	0.745159	.6.10.	0.790476	.2.4.6.	0.790476
.2.	0.706038	.6.11.	0.790476	.2.4.9.	0.755914
.3.	0.745113	.9.10.	0.775777	.2.4.10.	0.775777
.4.	0.751817	.9.11.	0.755914	.2.4.11.	0.75334
.5.	0.798698	.10.11.	0.775777	.2.5.6.	0.798698
.6.	0.790476	.1.2.3.	0.745159	.2.5.9.	0.798698
.9.	0.755914	.1.2.4.	0.751817	.2.5.10.	0.798698
.10.	0.775777	.1.2.5.	0.798698	.2.5.11.	0.798698
.11.	0.75334	.1.2.6.	0.790476	.2.6.9.	0.790476
.1.2.	0.745159	.1.2.9.	0.755914	.2.6.10.	0.790476
.1.3.	0.745159	.1.2.10.	0.775777	.2.6.11.	0.790476
.1.4.	0.751817	.1.2.11.	0.75334	.2.9.10.	0.775777
.1.5.	0.798698	.1.3.4.	0.751817	.2.9.11.	0.755914
.1.6.	0.790476	.1.3.5.	0.798698	.2.10.11.	0.775777
.1.9.	0.755914	.1.3.6.	0.790476	.3.4.5.	0.798698
.1.10.	0.775777	.1.3.9.	0.755914	.3.4.6.	0.790476
.1.11.	0.75334	.1.3.10.	0.775777	.3.4.9.	0.755914
.2.3.	0.745113	.1.3.11.	0.75334	.3.4.10.	0.775777
.2.4.	0.751817	.1.4.5.	0.798698	.3.4.11.	0.75334
.2.5.	0.798698	.1.4.6.	0.790476	.3.5.6.	0.798698
.2.6.	0.790476	.1.4.9.	0.755914	.3.5.9.	0.798698
.2.9.	0.755914	.1.4.10.	0.775777	.3.5.10.	0.798698
.2.10.	0.775777	.1.4.11.	0.75334	.3.5.11.	0.798698
.2.11.	0.75334	.1.5.6.	0.798698	.3.6.9.	0.790476
.3.4.	0.751817	.1.5.9.	0.798698	.3.6.10.	0.790476

Table C.1: Continued

Stage3 (cont.)	
.3.6.11.	0.790476
.3.9.10.	0.775777
.3.9.11.	0.755914
.3.10.11.	0.775777
.4.5.6.	0.798698
.4.5.9.	0.798698
.4.5.10.	0.798698
.4.5.11.	0.798698
.4.6.9.	0.790476
.4.6.10.	0.790476
.4.6.11.	0.790476
.4.9.10.	0.775777
.4.9.11.	0.755914
.4.10.11.	0.775777
.5.6.9.	0.798698
.5.6.10.	0.798698
.5.6.11.	0.798698
.5.9.10.	0.798698
.5.9.11.	0.798698
.5.10.11.	0.798698
.6.9.10.	0.790476
.6.9.11.	0.790476
.6.10.11.	0.790476
.9.10.11.	0.775777
.1.2.3.4.	0.751817
.1.2.3.5.	0.798698
.1.2.3.6.	0.790476
.1.2.3.9.	0.755914
.1.2.3.10.	0.775777
.1.2.3.11.	0.75334
.1.2.4.5.	0.798698
.1.2.4.6.	0.790476
.1.2.4.9.	0.755914
.1.2.4.10.	0.775777
.1.2.4.11.	0.75334
.1.2.5.6.	0.798698
.1.2.5.9.	0.798698
.1.2.5.10.	0.798698
.1.2.5.11.	0.798698
.1.2.6.9.	0.790476
.1.2.6.10.	0.790476
.1.2.6.11.	0.790476
.1.2.9.10.	0.775777
.1.2.9.11.	0.755914
.1.2.10.11.	0.775777
.1.3.4.5.	0.798698
.1.3.4.6.	0.790476
.1.3.4.9.	0.755914

Stage3 (cont.)	
.1.3.4.10.	0.775777
.1.3.4.11.	0.75334
.1.3.5.6.	0.798698
.1.3.5.9.	0.798698
.1.3.5.10.	0.798698
.1.3.5.11.	0.798698
.1.3.6.9.	0.790476
.1.3.6.10.	0.790476
.1.3.6.11.	0.790476
.1.3.9.10.	0.775777
.1.3.9.11.	0.755914
.1.3.10.11.	0.775777
.1.4.5.6.	0.798698
.1.4.5.9.	0.798698
.1.4.5.10.	0.798698
.1.4.5.11.	0.798698
.1.4.6.9.	0.790476
.1.4.6.10.	0.790476
.1.4.6.11.	0.790476
.1.4.9.10.	0.775777
.1.4.9.11.	0.755914
.1.4.10.11.	0.775777
.1.5.6.9.	0.798698
.1.5.6.10.	0.798698
.1.5.6.11.	0.798698
.1.5.9.10.	0.798698
.1.5.9.11.	0.798698
.1.5.10.11.	0.798698
.1.6.9.10.	0.790476
.1.6.9.11.	0.790476
.1.6.10.11.	0.790476
.1.9.10.11.	0.775777
.2.3.4.5.	0.798698
.2.3.4.6.	0.790476
.2.3.4.9.	0.755914
.2.3.4.10.	0.775777
.2.3.4.11.	0.75334
.2.3.5.6.	0.790476
.2.3.5.9.	0.755914
.2.3.5.10.	0.775777
.2.3.5.11.	0.75334
.2.3.6.9.	0.790476
.2.3.6.10.	0.790476
.2.3.6.11.	0.790476
.2.3.9.10.	0.775777
.2.3.9.11.	0.755914
.2.3.10.11.	0.775777
.2.4.5.6.	0.798698
.2.4.5.9.	0.798698

Stage3 (cont.)	
.2.4.5.10.	0.798698
.2.4.5.11.	0.798698
.2.4.6.9.	0.790476
.2.4.6.10.	0.790476
.2.4.6.11.	0.790476
.2.4.9.10.	0.775777
.2.4.9.11.	0.755914
.2.4.10.11.	0.775777
.2.5.6.9.	0.798698
.2.5.6.10.	0.798698
.2.5.6.11.	0.798698
.2.5.9.10.	0.798698
.2.5.9.11.	0.798698
.2.5.10.11.	0.798698
.2.6.9.10.	0.790476
.2.6.9.11.	0.790476
.2.6.10.11.	0.790476
.2.9.10.11.	0.775777
.3.4.5.6.	0.798698
.3.4.5.9.	0.798698
.3.4.5.10.	0.798698
.3.4.5.11.	0.798698
.3.4.6.9.	0.790476
.3.4.6.10.	0.790476
.3.4.6.11.	0.790476
.3.4.9.10.	0.775777
.3.4.9.11.	0.755914
.3.4.10.11.	0.775777
.3.5.6.9.	0.798698
.3.5.6.10.	0.798698
.3.5.6.11.	0.798698
.3.5.9.10.	0.798698
.3.5.9.11.	0.798698
.3.5.10.11.	0.798698
.3.6.9.10.	0.790476
.3.6.9.11.	0.790476
.3.6.10.11.	0.790476
.3.9.10.11.	0.775777
.4.5.6.9.	0.798698
.4.5.6.10.	0.798698
.4.5.6.11.	0.798698
.4.5.9.10.	0.798698
.4.5.9.11.	0.798698
.4.5.10.11.	0.798698
.4.6.9.10.	0.790476
.4.6.9.11.	0.790476
.4.6.10.11.	0.790476
.4.9.10.11.	0.775777
.5.6.9.10.	0.798698

Table C.1: Continued

Stage3 (cont.)	
.5.6.9.11.	0.798698
.5.6.10.11.	0.798698
.5.9.10.11.	0.798698
.6.9.10.11.	0.790476
.1.2.3.4.5.	0.798698
.1.2.3.4.6.	0.790476
.1.2.3.4.9.	0.755914
.1.2.3.4.10.	0.775777
.1.2.3.4.11.	0.75334
.1.2.3.5.6.	0.798698
.1.2.3.5.9.	0.798698
.1.2.3.5.10.	0.798698
.1.2.3.5.11.	0.798698
.1.2.3.6.9.	0.790476
.1.2.3.6.10.	0.790476
.1.2.3.6.11.	0.790476
.1.2.3.9.10.	0.775777
.1.2.3.9.11.	0.755914
.1.2.3.10.11.	0.775777
.1.2.4.5.6.	0.798698
.1.2.4.5.9.	0.798698
.1.2.4.5.10.	0.798698
.1.2.4.5.11.	0.798698
.1.2.4.6.9.	0.790476
.1.2.4.6.10.	0.790476
.1.2.4.6.11.	0.790476
.1.2.4.9.10.	0.775777
.1.2.4.9.11.	0.755914
.1.2.4.10.11.	0.775777
.1.2.5.6.9.	0.798698
.1.2.5.6.10.	0.798698
.1.2.5.6.11.	0.798698
.1.2.5.9.10.	0.798698
.1.2.5.9.11.	0.798698
.1.2.5.10.11.	0.798698
.1.2.6.9.10.	0.790476
.1.2.6.9.11.	0.790476
.1.2.6.10.11.	0.790476
.1.2.9.10.11.	0.775777
.1.3.4.5.6.	0.798698
.1.3.4.5.9.	0.798698
.1.3.4.5.10.	0.798698
.1.3.4.5.11.	0.798698
.1.3.4.6.9.	0.790476
.1.3.4.6.10.	0.790476
.1.3.4.6.11.	0.790476
.1.3.4.9.10.	0.790476
.1.3.4.9.11.	0.790476

Stage3 (cont.)	
.1.3.4.10.11.	0.775777
.1.3.5.6.9.	0.798698
.1.3.5.6.10.	0.798698
.1.3.5.6.11.	0.798698
.1.3.5.9.10.	0.798698
.1.3.5.9.11.	0.798698
.1.3.5.10.11.	0.798698
.1.3.6.9.10.	0.790476
.1.3.6.9.11.	0.790476
.1.3.6.10.11.	0.790476
.1.3.9.10.11.	0.775777
.1.4.5.6.9.	0.798698
.1.4.5.6.10.	0.798698
.1.4.5.6.11.	0.798698
.1.4.5.9.10.	0.798698
.1.4.5.9.11.	0.798698
.1.4.5.10.11.	0.798698
.1.4.6.9.10.	0.790476
.1.4.6.9.11.	0.790476
.1.4.6.10.11.	0.790476
.1.4.9.10.11.	0.775777
.1.5.6.9.10.	0.798698
.1.5.6.9.11.	0.798698
.1.5.6.10.11.	0.798698
.1.5.9.10.11.	0.798698
.1.6.9.10.11.	0.790476
.2.3.4.5.6.	0.798698
.2.3.4.5.9.	0.798698
.2.3.4.5.10.	0.798698
.2.3.4.5.11.	0.798698
.2.3.4.6.9.	0.790476
.2.3.4.6.10.	0.775777
.2.3.4.6.11.	0.775777
.2.3.4.9.10.	0.775777
.2.3.4.9.11.	0.755914
.2.3.4.10.11.	0.775777
.2.3.5.6.9.	0.798698
.2.3.5.6.10.	0.798698
.2.3.5.6.11.	0.798698
.2.3.5.9.10.	0.798698
.2.3.5.9.11.	0.798698
.2.3.5.10.11.	0.798698
.2.3.6.9.10.	0.790476
.2.3.6.9.11.	0.790476
.2.3.6.10.11.	0.790476
.2.3.9.10.11.	0.775777
.2.4.5.6.9.	0.798698
.2.4.5.6.10.	0.798698
.2.4.5.6.11.	0.798698

Stage3 (cont.)	
.2.4.5.9.10.	0.798698
.2.4.5.9.11.	0.798698
.2.4.5.10.11.	0.798698
.2.4.6.9.10.	0.790476
.2.4.6.9.11.	0.790476
.2.4.6.10.11.	0.790476
.2.4.9.10.11.	0.775777
.2.5.6.9.10.	0.798698
.2.5.6.9.11.	0.798698
.2.5.6.10.11.	0.798698
.2.5.9.10.11.	0.798698
.2.6.9.10.11.	0.790476
.3.4.5.6.9.	0.798698
.3.4.5.6.10.	0.798698
.3.4.5.6.11.	0.798698
.3.4.5.9.10.	0.798698
.3.4.5.9.11.	0.798698
.3.4.5.10.11.	0.798698
.3.4.6.9.10.	0.790476
.3.4.6.9.11.	0.790476
.3.4.6.10.11.	0.790476
.3.4.9.10.11.	0.775777
.3.5.6.9.10.	0.798698
.3.5.6.9.11.	0.798698
.3.5.6.10.11.	0.798698
.3.5.9.10.11.	0.798698
.3.6.9.10.11.	0.790476
.4.5.6.9.10.	0.798698
.4.5.6.9.11.	0.798698
.4.5.6.10.11.	0.798698
.4.5.9.10.11.	0.798698
.4.6.9.10.11.	0.790476
.5.6.9.10.11.	0.798698
.1.2.3.4.5.6.	0.798698
.1.2.3.4.5.9.	0.798698
.1.2.3.4.5.10.	0.798698
.1.2.3.4.5.11.	0.798698
.1.2.3.4.6.9.	0.790476
.1.2.3.4.6.10.	0.790476
.1.2.3.4.6.11.	0.790476
.1.2.3.4.9.10.	0.775777
.1.2.3.4.9.11.	0.755914
.1.2.3.4.10.11.	0.775777
.1.2.3.5.6.9.	0.798698
.1.2.3.5.6.10.	0.798698
.1.2.3.5.6.11.	0.798698
.1.2.3.5.9.10.	0.798698
.1.2.3.5.9.11.	0.798698
.1.2.3.5.10.11.	0.798698

Table C.1: Continued

Stage3 (cont.)	
.1.2.3.6.9.10.	0.790476
.1.2.3.6.9.11.	0.790476
.1.2.3.6.10.11.	0.790476
.1.2.3.9.10.11.	0.775777
.1.2.4.5.6.9.	0.798698
.1.2.4.5.6.10.	0.798698
.1.2.4.5.6.11.	0.798698
.1.2.4.5.9.10.	0.798698
.1.2.4.5.9.11.	0.798698
.1.2.4.5.10.11.	0.798698
.1.2.4.6.9.10.	0.775777
.1.2.4.6.9.11.	0.755914
.1.2.4.6.10.11.	0.790476
.1.2.4.9.10.11.	0.775777
.1.2.5.6.9.10.	0.798698
.1.2.5.6.9.11.	0.798698
.1.2.5.6.10.11.	0.798698
.1.2.5.9.10.11.	0.798698
.1.2.6.9.10.11.	0.790476
.1.3.4.5.6.9.	0.798698
.1.3.4.5.6.10.	0.798698
.1.3.4.5.6.11.	0.798698
.1.3.4.5.9.10.	0.798698
.1.3.4.5.9.11.	0.798698
.1.3.4.5.10.11.	0.798698
.1.3.4.6.9.10.	0.790476
.1.3.4.6.9.11.	0.790476
.1.3.4.6.10.11.	0.790476
.1.3.4.9.10.11.	0.775777
.1.3.5.6.9.10.	0.798698
.1.3.5.6.9.11.	0.798698
.1.3.5.6.10.11.	0.798698
.1.3.5.9.10.11.	0.798698
.1.3.6.9.10.11.	0.790476
.1.4.5.6.9.10.	0.798698
.1.4.5.6.9.11.	0.798698
.1.4.5.6.10.11.	0.798698
.1.4.5.9.10.11.	0.798698
.1.4.6.9.10.11.	0.790476
.1.5.6.9.10.11.	0.798698
.2.3.4.5.6.9.	0.798698
.2.3.4.5.6.10.	0.798698
.2.3.4.5.6.11.	0.798698
.2.3.4.5.9.10.	0.798698
.2.3.4.5.9.11.	0.798698
.2.3.4.5.10.11.	0.798698
.2.3.4.6.9.10.	0.790476
.2.3.4.6.9.11.	0.790476

Stage3 (cont.)	
.2.3.4.6.10.11.	0.790476
.2.3.4.9.10.11.	0.775777
.2.3.5.6.9.10.	0.798698
.2.3.5.6.9.11.	0.798698
.2.3.5.6.10.11.	0.798698
.2.3.5.9.10.11.	0.798698
.2.3.6.9.10.11.	0.790476
.2.4.5.6.9.10.	0.798698
.2.4.5.6.9.11.	0.798698
.2.4.5.6.10.11.	0.798698
.2.4.5.9.10.11.	0.798698
.2.4.6.9.10.11.	0.790476
.2.5.6.9.10.11.	0.798698
.3.4.5.6.9.10.	0.798698
.3.4.5.6.9.11.	0.798698
.3.4.5.6.10.11.	0.798698
.3.4.5.9.10.11.	0.798698
.3.4.6.9.10.11.	0.790476
.3.5.6.9.10.11.	0.798698
.4.5.6.9.10.11.	0.798698
.1.2.3.4.5.6.9.	0.798698
.1.2.3.4.5.6.10.	0.798698
.1.2.3.4.5.6.11.	0.798698
.1.2.3.4.5.9.10.	0.798698
.1.2.3.4.5.9.11.	0.798698
.1.2.3.4.5.10.11.	0.798698
.1.2.3.4.6.9.10.	0.790476
.1.2.3.4.6.9.11.	0.790476
.1.2.3.4.6.10.11.	0.790476
.1.2.3.4.9.10.11.	0.775777
.1.2.3.5.6.9.10.	0.798698
.1.2.3.5.6.9.11.	0.798698
.1.2.3.5.6.10.11.	0.798698
.1.2.3.5.9.10.11.	0.798698
.1.2.3.6.9.10.11.	0.790476
.1.2.4.5.6.9.10.	0.798698
.1.2.4.5.6.9.11.	0.798698
.1.2.4.5.6.10.11.	0.798698
.1.2.4.5.9.10.11.	0.798698
.1.2.4.6.9.10.11.	0.790476
.1.2.5.6.9.10.11.	0.798698
.1.3.4.5.6.9.10.	0.798698
.1.3.4.5.6.9.11.	0.798698
.1.3.4.5.6.10.11.	0.798698
.1.3.4.5.9.10.11.	0.798698
.1.3.4.6.9.10.11.	0.790476
.1.3.5.6.9.10.11.	0.798698
.1.4.5.6.9.10.11.	0.798698
.2.3.4.5.6.9.10.	0.798698

Stage3 (cont.)	
.2.3.4.5.6.9.11.	0.798698
.2.3.4.5.6.10.11.	0.798698
.2.3.4.5.9.10.11.	0.798698
.2.3.4.6.9.10.11.	0.790476
.2.3.5.6.9.10.11.	0.798698
.2.4.5.6.9.10.11.	0.798698
.3.4.5.6.9.10.11.	0.798698
.1.2.3.4.5.6.9.10.	0.798698
.1.2.3.4.5.6.9.11.	0.798698
.1.2.3.4.5.6.10.11.	0.798698
.1.2.3.4.5.9.10.11.	0.798698
.1.2.3.4.6.9.10.11.	0.790476
.1.2.3.5.6.9.10.11.	0.798698
.1.2.4.5.6.9.10.11.	0.798698
.1.3.4.5.6.9.10.11.	0.798698
.2.3.4.5.6.9.10.11.	0.798698
.1.2.3.4.5.6.9.10.11.	0.798698
18	
.1.	0.710457
.2.	0.786788
.3.	0.787223
.4.	0.731027
.5.	0.781668
.6.	0.702819
.9.	0.719458
.10.	0.780905
.11.	0.740472
.1.2.	0.786788
.1.3.	0.787223
.1.4.	0.731027
.1.5.	0.781668
.1.6.	0.710457
.1.9.	0.719458
.1.10.	0.780905
.1.11.	0.740472
.2.3.	0.787223
.2.4.	0.786788
.2.5.	0.786788
.2.6.	0.786788
.2.9.	0.786788
.2.10.	0.786788
.2.11.	0.786788
.3.4.	0.787223
.3.5.	0.787223
.3.6.	0.787223
.3.9.	0.787223
.3.10.	0.787223
.3.11.	0.787223
.4.5.	0.781668

Table C.1: Continued

Stage3 (cont.)	
.4.6.	0.731027
.4.9.	0.731027
.4.10.	0.780905
.4.11.	0.740472
.5.6.	0.781668
.5.9.	0.781668
.5.10.	0.781668
.5.11.	0.781668
.6.9.	0.719458
.6.10.	0.780905
.6.11.	0.740472
.9.10.	0.780905
.9.11.	0.740472
.10.11.	0.780905
.1.2.3.	0.787223
.1.2.4.	0.786788
.1.2.5.	0.786788
.1.2.6.	0.786788
.1.2.9.	0.786788
.1.2.10.	0.786788
.1.2.11.	0.786788
.1.3.4.	0.787223
.1.3.5.	0.787223
.1.3.6.	0.787223
.1.3.9.	0.787223
.1.3.10.	0.787223
.1.3.11.	0.787223
.1.4.5.	0.781668
.1.4.6.	0.731027
.1.4.9.	0.731027
.1.4.10.	0.780905
.1.4.11.	0.740472
.1.5.6.	0.781668
.1.5.9.	0.781668
.1.5.10.	0.781668
.1.5.11.	0.781668
.1.6.9.	0.719458
.1.6.10.	0.780905
.1.6.11.	0.740472
.1.9.10.	0.780905
.1.9.11.	0.740472
.1.10.11.	0.780905
.2.3.4.	0.787223
.2.3.5.	0.787223
.2.3.6.	0.787223
.2.3.9.	0.787223
.2.3.10.	0.787223
.2.3.11.	0.787223

Stage3 (cont.)	
.2.4.5.	0.786788
.2.4.6.	0.786788
.2.4.9.	0.786788
.2.4.10.	0.786788
.2.4.11.	0.786788
.2.5.6.	0.786788
.2.5.9.	0.786788
.2.5.10.	0.786788
.2.5.11.	0.786788
.2.6.9.	0.786788
.2.6.10.	0.786788
.2.6.11.	0.786788
.2.9.10.	0.786788
.2.9.11.	0.786788
.2.10.11.	0.786788
.3.4.5.	0.787223
.3.4.6.	0.787223
.3.4.9.	0.787223
.3.4.10.	0.787223
.3.4.11.	0.787223
.3.5.6.	0.787223
.3.5.9.	0.787223
.3.5.10.	0.787223
.3.5.11.	0.787223
.3.6.9.	0.787223
.3.6.10.	0.787223
.3.6.11.	0.787223
.3.9.10.	0.787223
.3.9.11.	0.787223
.3.10.11.	0.787223
.4.5.6.	0.781668
.4.5.9.	0.781668
.4.5.10.	0.781668
.4.5.11.	0.781668
.4.6.9.	0.731027
.4.6.10.	0.780905
.4.6.11.	0.740472
.4.9.10.	0.780905
.4.9.11.	0.740472
.4.10.11.	0.780905
.5.6.9.	0.781668
.5.6.10.	0.781668
.5.6.11.	0.781668
.5.9.10.	0.781668
.5.9.11.	0.781668
.5.10.11.	0.781668
.6.9.10.	0.780905
.6.9.11.	0.740472
.6.10.11.	0.780905

Stage3 (cont.)	
.9.10.11.	0.780905
.1.2.3.4.	0.787223
.1.2.3.5.	0.787223
.1.2.3.6.	0.787223
.1.2.3.9.	0.787223
.1.2.3.10.	0.787223
.1.2.3.11.	0.787223
.1.2.4.5.	0.786788
.1.2.4.6.	0.786788
.1.2.4.9.	0.786788
.1.2.4.10.	0.786788
.1.2.4.11.	0.786788
.1.2.5.6.	0.786788
.1.2.5.9.	0.786788
.1.2.5.10.	0.786788
.1.2.5.11.	0.786788
.1.2.6.9.	0.786788
.1.2.6.10.	0.786788
.1.2.6.11.	0.786788
.1.2.9.10.	0.786788
.1.2.9.11.	0.786788
.1.2.10.11.	0.786788
.1.3.4.5.	0.787223
.1.3.4.6.	0.787223
.1.3.4.9.	0.787223
.1.3.4.10.	0.787223
.1.3.4.11.	0.787223
.1.3.5.6.	0.787223
.1.3.5.9.	0.787223
.1.3.5.10.	0.787223
.1.3.5.11.	0.787223
.1.3.6.9.	0.787223
.1.3.6.10.	0.787223
.1.3.6.11.	0.787223
.1.3.9.10.	0.787223
.1.3.9.11.	0.787223
.1.3.10.11.	0.787223
.1.4.5.6.	0.781668
.1.4.5.9.	0.781668
.1.4.5.10.	0.781668
.1.4.5.11.	0.781668
.1.4.6.9.	0.731027
.1.4.6.10.	0.780905
.1.4.6.11.	0.740472
.1.4.9.10.	0.780905
.1.4.9.11.	0.740472
.1.4.10.11.	0.780905
.1.5.6.9.	0.781668
.1.5.6.10.	0.781668

Table C.1: Continued

Stage3 (cont.)	
.1.5.6.11.	0.781668
.1.5.9.10.	0.781668
.1.5.9.11.	0.781668
.1.5.10.11.	0.781668
.1.6.9.10.	0.780905
.1.6.9.11.	0.740472
.1.6.10.11.	0.780905
.1.9.10.11.	0.780905
.2.3.4.5.	0.787223
.2.3.4.6.	0.787223
.2.3.4.9.	0.787223
.2.3.4.10.	0.787223
.2.3.4.11.	0.787223
.2.3.5.6.	0.787223
.2.3.5.9.	0.787223
.2.3.5.10.	0.787223
.2.3.5.11.	0.787223
.2.3.6.9.	0.787223
.2.3.6.10.	0.787223
.2.3.6.11.	0.787223
.2.3.9.10.	0.787223
.2.3.9.11.	0.787223
.2.3.10.11.	0.787223
.2.4.5.6.	0.786788
.2.4.5.9.	0.786788
.2.4.5.10.	0.786788
.2.4.5.11.	0.786788
.2.4.6.9.	0.786788
.2.4.6.10.	0.786788
.2.4.6.11.	0.786788
.2.4.9.10.	0.786788
.2.4.9.11.	0.786788
.2.4.10.11.	0.786788
.2.5.6.9.	0.786788
.2.5.6.10.	0.786788
.2.5.6.11.	0.786788
.2.5.9.10.	0.786788
.2.5.9.11.	0.786788
.2.5.10.11.	0.786788
.2.6.9.10.	0.786788
.2.6.9.11.	0.786788
.2.6.10.11.	0.786788
.2.9.10.11.	0.786788
.3.4.5.6.	0.787223
.3.4.5.9.	0.787223
.3.4.5.10.	0.787223
.3.4.5.11.	0.787223
.3.4.6.9.	0.787223

Stage3 (cont.)	
.3.4.6.10.	0.787223
.3.4.6.11.	0.787223
.3.4.9.10.	0.787223
.3.4.9.11.	0.787223
.3.4.10.11.	0.787223
.3.5.6.9.	0.787223
.3.5.6.10.	0.787223
.3.5.6.11.	0.787223
.3.5.9.10.	0.787223
.3.5.9.11.	0.787223
.3.5.10.11.	0.787223
.3.6.9.10.	0.787223
.3.6.9.11.	0.787223
.3.6.10.11.	0.787223
.3.9.10.11.	0.787223
.4.5.6.9.	0.781668
.4.5.6.10.	0.781668
.4.5.6.11.	0.781668
.4.5.9.10.	0.781668
.4.5.9.11.	0.781668
.4.5.10.11.	0.781668
.4.6.9.10.	0.780905
.4.6.9.11.	0.740472
.4.6.10.11.	0.780905
.4.9.10.11.	0.780905
.5.6.9.10.	0.781668
.5.6.9.11.	0.781668
.5.6.10.11.	0.781668
.5.9.10.11.	0.781668
.6.9.10.11.	0.780905
.1.2.3.4.5.	0.787223
.1.2.3.4.6.	0.787223
.1.2.3.4.9.	0.787223
.1.2.3.4.10.	0.787223
.1.2.3.4.11.	0.787223
.1.2.3.5.6.	0.787223
.1.2.3.5.9.	0.787223
.1.2.3.5.10.	0.787223
.1.2.3.5.11.	0.787223
.1.2.3.6.9.	0.787223
.1.2.3.6.10.	0.787223
.1.2.3.6.11.	0.787223
.1.2.3.9.10.	0.787223
.1.2.3.9.11.	0.787223
.1.2.3.10.11.	0.787223
.1.2.4.5.6.	0.786788
.1.2.4.5.9.	0.786788
.1.2.4.5.10.	0.786788
.1.2.4.5.11.	0.786788

Stage3 (cont.)	
.1.2.4.6.9.	0.786788
.1.2.4.6.10.	0.786788
.1.2.4.6.11.	0.786788
.1.2.4.9.10.	0.786788
.1.2.4.9.11.	0.786788
.1.2.4.10.11.	0.786788
.1.2.5.6.9.	0.786788
.1.2.5.6.10.	0.786788
.1.2.5.6.11.	0.786788
.1.2.5.9.10.	0.786788
.1.2.5.9.11.	0.786788
.1.2.5.10.11.	0.786788
.1.2.6.9.10.	0.786788
.1.2.6.9.11.	0.786788
.1.2.6.10.11.	0.786788
.1.2.9.10.11.	0.786788
.1.3.4.5.6.	0.787223
.1.3.4.5.9.	0.787223
.1.3.4.5.10.	0.787223
.1.3.4.5.11.	0.787223
.1.3.4.6.9.	0.787223
.1.3.4.6.10.	0.787223
.1.3.4.6.11.	0.787223
.1.3.4.9.10.	0.787223
.1.3.4.9.11.	0.787223
.1.3.4.10.11.	0.787223
.1.3.5.6.9.	0.787223
.1.3.5.6.10.	0.787223
.1.3.5.6.11.	0.787223
.1.3.5.9.10.	0.787223
.1.3.5.9.11.	0.787223
.1.3.5.10.11.	0.787223
.1.3.6.9.10.	0.787223
.1.3.6.9.11.	0.787223
.1.3.6.10.11.	0.787223
.1.3.9.10.11.	0.787223
.1.4.5.6.9.	0.781668
.1.4.5.6.10.	0.781668
.1.4.5.6.11.	0.781668
.1.4.5.9.10.	0.781668
.1.4.5.9.11.	0.781668
.1.4.5.10.11.	0.781668
.1.4.6.9.10.	0.780905
.1.4.6.9.11.	0.740472
.1.4.6.10.11.	0.780905
.1.4.9.10.11.	0.780905
.1.5.6.9.10.	0.781668
.1.5.6.9.11.	0.781668
.1.5.6.10.11.	0.781668

Table C.1: Continued

Stage3 (cont.)	
.1.5.9.10.11.	0.781668
.1.6.9.10.11.	0.780905
.2.3.4.5.6.	0.787223
.2.3.4.5.9.	0.787223
.2.3.4.5.10.	0.787223
.2.3.4.5.11.	0.787223
.2.3.4.6.9.	0.787223
.2.3.4.6.10.	0.787223
.2.3.4.6.11.	0.787223
.2.3.4.9.10.	0.787223
.2.3.4.9.11.	0.787223
.2.3.4.10.11.	0.787223
.2.3.5.6.9.	0.787223
.2.3.5.6.10.	0.787223
.2.3.5.6.11.	0.787223
.2.3.5.9.10.	0.787223
.2.3.5.9.11.	0.787223
.2.3.5.10.11.	0.787223
.2.3.6.9.10.	0.787223
.2.3.6.9.11.	0.787223
.2.3.6.10.11.	0.787223
.2.3.9.10.11.	0.787223
.2.4.5.6.9.	0.786788
.2.4.5.6.10.	0.786788
.2.4.5.6.11.	0.786788
.2.4.5.9.10.	0.786788
.2.4.5.9.11.	0.786788
.2.4.5.10.11.	0.786788
.2.4.6.9.10.	0.786788
.2.4.6.9.11.	0.786788
.2.4.6.10.11.	0.786788
.2.4.9.10.11.	0.786788
.2.5.6.9.10.	0.786788
.2.5.6.9.11.	0.786788
.2.5.6.10.11.	0.786788
.2.5.9.10.11.	0.786788
.2.6.9.10.11.	0.786788
.3.4.5.6.9.	0.787223
.3.4.5.6.10.	0.787223
.3.4.5.6.11.	0.787223
.3.4.5.9.10.	0.787223
.3.4.5.9.11.	0.787223
.3.4.5.10.11.	0.787223
.3.4.6.9.10.	0.787223
.3.4.6.9.11.	0.787223
.3.4.6.10.11.	0.787223
.3.4.9.10.11.	0.787223
.3.5.6.9.10.	0.787223

Stage3 (cont.)	
.3.5.6.9.11.	0.787223
.3.5.6.10.11.	0.787223
.3.5.9.10.11.	0.787223
.3.6.9.10.11.	0.787223
.4.5.6.9.10.	0.781668
.4.5.6.9.11.	0.781668
.4.5.6.10.11.	0.781668
.4.5.9.10.11.	0.781668
.4.6.9.10.11.	0.780905
.5.6.9.10.11.	0.781668
.1.2.3.4.5.6.	0.787223
.1.2.3.4.5.9.	0.787223
.1.2.3.4.5.10.	0.787223
.1.2.3.4.5.11.	0.787223
.1.2.3.4.6.9.	0.787223
.1.2.3.4.6.10.	0.787223
.1.2.3.4.6.11.	0.787223
.1.2.3.4.9.10.	0.787223
.1.2.3.4.9.11.	0.787223
.1.2.3.4.10.11.	0.787223
.1.2.3.5.6.9.	0.787223
.1.2.3.5.6.10.	0.787223
.1.2.3.5.6.11.	0.787223
.1.2.3.5.9.10.	0.787223
.1.2.3.5.9.11.	0.787223
.1.2.3.5.10.11.	0.787223
.1.2.3.6.9.10.	0.787223
.1.2.3.6.9.11.	0.787223
.1.2.3.6.10.11.	0.787223
.1.2.3.9.10.11.	0.787223
.1.2.4.5.6.9.	0.786788
.1.2.4.5.6.10.	0.786788
.1.2.4.5.6.11.	0.786788
.1.2.4.5.9.10.	0.786788
.1.2.4.5.9.11.	0.786788
.1.2.4.5.10.11.	0.786788
.1.2.4.6.9.10.	0.786788
.1.2.4.6.9.11.	0.786788
.1.2.4.6.10.11.	0.786788
.1.2.4.9.10.11.	0.786788
.1.2.5.6.9.10.	0.786788
.1.2.5.6.9.11.	0.786788
.1.2.5.6.10.11.	0.786788
.1.2.5.9.10.11.	0.786788
.1.2.6.9.10.11.	0.786788
.1.3.4.5.6.9.	0.787223
.1.3.4.5.6.10.	0.787223
.1.3.4.5.6.11.	0.787223
.1.3.4.5.9.10.	0.787223

Stage3 (cont.)	
.1.3.4.5.9.11.	0.787223
.1.3.4.5.10.11.	0.787223
.1.3.4.6.9.10.	0.787223
.1.3.4.6.9.11.	0.787223
.1.3.4.6.10.11.	0.787223
.1.3.4.9.10.11.	0.787223
.1.3.5.6.9.10.	0.787223
.1.3.5.6.9.11.	0.787223
.1.3.5.6.10.11.	0.787223
.1.3.5.9.10.11.	0.787223
.1.3.6.9.10.11.	0.787223
.1.4.5.6.9.10.	0.781668
.1.4.5.6.9.11.	0.781668
.1.4.5.6.10.11.	0.781668
.1.4.5.9.10.11.	0.781668
.1.4.6.9.10.11.	0.780905
.1.5.6.9.10.11.	0.781668
.2.3.4.5.6.9.	0.787223
.2.3.4.5.6.10.	0.787223
.2.3.4.5.6.11.	0.787223
.2.3.4.5.9.10.	0.787223
.2.3.4.5.9.11.	0.787223
.2.3.4.5.10.11.	0.787223
.2.3.4.6.9.10.	0.787223
.2.3.4.6.9.11.	0.787223
.2.3.4.6.10.11.	0.787223
.2.3.4.9.10.11.	0.787223
.2.3.5.6.9.10.	0.787223
.2.3.5.6.9.11.	0.787223
.2.3.5.6.10.11.	0.787223
.2.3.5.9.10.11.	0.787223
.2.3.6.9.10.11.	0.787223
.2.4.5.6.9.10.	0.786788
.2.4.5.6.9.11.	0.786788
.2.4.5.6.10.11.	0.786788
.2.4.5.9.10.11.	0.786788
.2.4.6.9.10.11.	0.786788
.2.5.6.9.10.11.	0.786788
.3.4.5.6.9.10.	0.787223
.3.4.5.6.9.11.	0.787223
.3.4.5.6.10.11.	0.787223
.3.4.5.9.10.11.	0.787223
.3.4.6.9.10.11.	0.787223
.3.5.6.9.10.11.	0.787223
.4.5.6.9.10.11.	0.781668
.1.2.3.4.5.6.9.	0.787223
.1.2.3.4.5.6.10.	0.787223
.1.2.3.4.5.6.11.	0.787223
.1.2.3.4.5.9.10.	0.787223

Table C.1: Continued

Stage3 (cont.)	
.1.2.3.4.5.9.11.	0.787223
.1.2.3.4.5.10.11.	0.787223
.1.2.3.4.6.9.10.	0.787223
.1.2.3.4.6.9.11.	0.787223
.1.2.3.4.6.10.11.	0.787223
.1.2.3.4.9.10.11.	0.787223
.1.2.3.5.6.9.10.	0.787223
.1.2.3.5.6.9.11.	0.787223
.1.2.3.5.6.10.11.	0.787223
.1.2.3.5.9.10.11.	0.787223
.1.2.3.6.9.10.11.	0.787223
.1.2.4.5.6.9.10.	0.786788
.1.2.4.5.6.9.11.	0.786788
.1.2.4.5.6.10.11.	0.786788
.1.2.4.5.9.10.11.	0.786788
.1.2.4.6.9.10.11.	0.786788
.1.2.5.6.9.10.11.	0.786788
.1.3.4.5.6.9.10.	0.787223
.1.3.4.5.6.9.11.	0.787223
.1.3.4.5.6.10.11.	0.787223
.1.3.4.5.9.10.11.	0.787223
.1.3.4.6.9.10.11.	0.787223
.1.3.5.6.9.10.11.	0.787223
.1.4.5.6.9.10.11.	0.781668
.2.3.4.5.6.9.10.	0.787223
.2.3.4.5.6.9.11.	0.787223
.2.3.4.5.6.10.11.	0.787223
.2.3.4.5.9.10.11.	0.787223
.2.3.4.6.9.10.11.	0.787223
.2.3.5.6.9.10.11.	0.787223
.2.4.5.6.9.10.11.	0.786788
.3.4.5.6.9.10.11.	0.787223
.1.2.3.4.5.6.9.10.	0.787223
.1.2.3.4.5.6.9.11.	0.787223
.1.2.3.4.5.6.10.11.	0.787223
.1.2.3.4.5.9.10.11.	0.787223
.1.2.3.4.6.9.10.11.	0.787223
.1.2.3.5.6.9.10.11.	0.787223
.1.2.4.5.6.9.10.11.	0.786788
.1.3.4.5.6.9.10.11.	0.787223
.2.3.4.5.6.9.10.11.	0.787223
.1.2.3.4.5.6.9.10.11.	0.787223
19	
.4.	0.71013
.5.	0.779123
.6.	0.785841
.4.5.	0.779123
.4.6.	0.785841

Stage3 (cont.)	
.5.6.	0.785841
.4.5.6.	0.785841
20	
.5.	0.726599
.6.	0.758447
.5.6.	0.758447
21	
.5.	0.774223
.6.	0.77184
.7.	0.727165
.8.	0.787903
.5.6.	0.774223
.5.7.	0.774223
.5.8.	0.787903
.6.7.	0.77184
.6.8.	0.787903
.7.8.	0.787903
.5.6.7.	0.774223
.5.6.8.	0.787903
.5.7.8.	0.787903
.6.7.8.	0.787903
.5.6.7.8.	0.787903
22	
.5.	0.776327
.8.	0.759945
.5.8.	0.776327
Stage4	
1	
.17.	0.979607
.18.	0.976998
.17.18.	0.979607
2	
.17.	0.908642
.18.	0.933632
.17.18.	0.933632
23	
.17.	0.734244
.18.	0.739838
.17.18.	0.739838
3	
.17.	0.978191
.18.	0.920969
.17.18.	0.978191
4	
.19.	0.749796
6	
.20.	0.750318
7	

Stage4 (cont.)	
.21.	0.705838
8	
.22.	0.712878
9	
.18.	0.960467
10	
.18.	0.99889
Stage5	
1	
.1.	0.943768
2	
.2.	0.993247
23	
.23.	0.779183
3	
.3.	0.902563
24	
.1.	0.754864
.2.	0.739963
.23.	0.742841
.3.	0.741467
.4.	0.770902
.6.	0.774288
.1.2.	0.754864
.1.23.	0.754864
.1.3.	0.754864
.1.4.	0.770902
.1.6.	0.774288
.2.23.	0.742841
.2.3.	0.741467
.2.4.	0.770902
.2.6.	0.774288
.23.3.	0.742841
.23.4.	0.770902
.23.6.	0.774288
.3.4.	0.770902
.3.6.	0.774288
.4.6.	0.774288
.1.2.23.	0.754864
.1.2.3.	0.754864
.1.2.4.	0.770902
.1.2.6.	0.774288
.1.23.3.	0.754864
.1.23.4.	0.770902
.1.23.6.	0.774288
.1.3.4.	0.770902
.1.3.6.	0.774288
.1.4.6.	0.774288

Table C.1: Continued

Stage5 (cont.)	
.2.23.3.	0.742841
.2.23.4.	0.770902
.2.23.6.	0.774288
.2.3.4.	0.770902
.2.3.6.	0.774288
.2.4.6.	0.774288
.23.3.4.	0.770902
.23.3.6.	0.774288
.23.4.6.	0.774288
.3.4.6.	0.774288
.1.2.23.3.	0.754864
.1.2.23.4.	0.770902
.1.2.23.6.	0.774288
.1.2.3.4.	0.770902
.1.2.3.6.	0.774288
.1.2.4.6.	0.774288
.1.23.3.4.	0.770902
.1.23.3.6.	0.774288
.1.23.4.6.	0.774288
.1.3.4.6.	0.774288
.2.23.3.4.	0.770902
.2.23.3.6.	0.774288
.2.23.4.6.	0.774288
.2.3.4.6.	0.774288
.23.3.4.6.	0.774288
.1.2.23.3.4.	0.770902
.1.2.23.3.6.	0.774288
.1.2.23.4.6.	0.774288
.1.2.3.4.6.	0.774288
.1.23.3.4.6.	0.774288
.2.23.3.4.6.	0.774288
.1.2.23.3.4.6.	0.774288
6	
.6.	0.797294
7	
.7.	0.708267
8	
.8.	0.766288
9	
.9.	0.972647
10	
.10.	0.93397
Stage6	
25	
.1.	0.799757
.2.	0.767152
.23.	0.718743

Stage6 (cont.)	
.3.	0.745825
.24.	0.779844
.9.	0.778076
.10.	0.717521
.1.2.	0.799757
.1.23.	0.799757
.1.3.	0.799757
.1.24.	0.799757
.1.9.	0.799757
.1.10.	0.799757
.2.23.	0.767152
.2.3.	0.767152
.2.24.	0.779844
.2.9.	0.778076
.2.10.	0.767152
.23.3.	0.745825
.23.24.	0.779844
.23.9.	0.778076
.23.10.	0.718743
.3.24.	0.779844
.3.9.	0.778076
.3.10.	0.745825
.24.9.	0.779844
.24.10.	0.779844
.9.10.	0.778076
.1.2.23.	0.799757
.1.2.3.	0.799757
.1.2.24.	0.799757
.1.2.9.	0.799757
.1.2.10.	0.799757
.1.23.3.	0.799757
.1.23.24.	0.799757
.1.23.9.	0.799757
.1.23.10.	0.799757
.1.3.24.	0.799757
.1.3.9.	0.799757
.1.3.10.	0.799757
.1.24.9.	0.799757
.1.24.10.	0.799757
.1.9.10.	0.799757
.2.23.3.	0.767152
.2.23.24.	0.779844
.2.23.9.	0.778076
.2.23.10.	0.767152
.2.3.24.	0.779844
.2.3.9.	0.778076
.2.3.10.	0.767152
.2.24.9.	0.779844
.2.24.10.	0.779844

Stage6 (cont.)	
.2.9.10.	0.778076
.23.3.24.	0.779844
.23.3.9.	0.778076
.23.3.10.	0.745825
.23.24.9.	0.779844
.23.24.10.	0.779844
.23.9.10.	0.778076
.3.24.9.	0.779844
.3.24.10.	0.779844
.3.9.10.	0.778076
.24.9.10.	0.779844
.1.2.23.3.	0.799757
.1.2.23.24.	0.799757
.1.2.23.9.	0.799757
.1.2.23.10.	0.799757
.1.2.3.24.	0.799757
.1.2.3.9.	0.799757
.1.2.3.10.	0.799757
.1.2.24.9.	0.799757
.1.2.24.10.	0.799757
.1.2.9.10.	0.799757
.1.23.3.24.	0.799757
.1.23.3.9.	0.799757
.1.23.3.10.	0.799757
.1.23.24.9.	0.799757
.1.23.24.10.	0.799757
.1.23.9.10.	0.799757
.1.3.24.9.	0.799757
.1.3.24.10.	0.799757
.1.3.9.10.	0.799757
.1.24.9.10.	0.799757
.2.23.3.24.	0.779844
.2.23.3.9.	0.778076
.2.23.3.10.	0.767152
.2.23.24.9.	0.779844
.2.23.24.10.	0.779844
.2.23.9.10.	0.778076
.2.3.24.9.	0.779844
.2.3.24.10.	0.779844
.2.3.9.10.	0.778076
.2.24.9.10.	0.779844
.23.3.24.9.	0.779844
.23.3.24.10.	0.779844
.23.3.9.10.	0.778076
.23.24.9.10.	0.779844
.3.24.9.10.	0.779844
.1.2.23.3.24.	0.799757
.1.2.23.3.9.	0.799757
.1.2.23.3.10.	0.799757

Table C.1: Continued

Stage6 (cont.)	
.1.2.23.24.9.	0.799757
.1.2.23.24.10.	0.799757
.1.2.23.9.10.	0.799757
.1.2.3.24.9.	0.799757
.1.2.3.24.10.	0.799757
.1.2.3.9.10.	0.799757
.1.2.24.9.10.	0.799757
.1.23.3.24.9.	0.799757
.1.23.3.24.10.	0.799757
.1.23.3.9.10.	0.799757
.1.23.24.9.10.	0.799757
.1.3.24.9.10.	0.799757
.2.23.3.24.9.	0.779844
.2.23.3.24.10.	0.779844
.2.23.3.9.10.	0.778076
.2.23.24.9.10.	0.779844
.2.3.24.9.10.	0.779844
.23.3.24.9.10.	0.779844
.1.2.23.3.24.9.	0.799757
.1.2.23.3.24.10.	0.799757
.1.2.23.3.9.10.	0.799757
.1.2.23.24.9.10.	0.799757
.1.2.3.24.9.10.	0.799757
.1.23.3.24.9.10.	0.799757
.2.23.3.24.9.10.	0.779844
.1.2.23.3.24.9.10.	0.799757
26	
.1.	0.740536
.2.	0.757362
.23.	0.760781
.3.	0.788609
.24.	0.705905
.9.	0.753247
.10.	0.766532
.1.2.	0.757362
.1.23.	0.760781
.1.3.	0.788609
.1.24.	0.740536
.1.9.	0.753247
.1.10.	0.766532
.2.23.	0.760781
.2.3.	0.788609
.2.24.	0.757362
.2.9.	0.757362
.2.10.	0.766532
.23.3.	0.788609
.23.24.	0.760781
.23.9.	0.760781

Stage6 (cont.)	
.23.10.	0.766532
.3.24.	0.788609
.3.9.	0.788609
.3.10.	0.788609
.24.9.	0.753247
.24.10.	0.766532
.9.10.	0.766532
.1.2.23.	0.760781
.1.2.3.	0.788609
.1.2.24.	0.757362
.1.2.9.	0.757362
.1.2.10.	0.766532
.1.23.3.	0.788609
.1.23.24.	0.760781
.1.23.9.	0.760781
.1.23.10.	0.766532
.1.3.24.	0.788609
.1.3.9.	0.788609
.1.3.10.	0.788609
.1.24.9.	0.753247
.1.24.10.	0.766532
.1.9.10.	0.766532
.2.23.3.	0.788609
.2.23.24.	0.760781
.2.23.9.	0.760781
.2.23.10.	0.766532
.2.3.24.	0.788609
.2.3.9.	0.788609
.2.3.10.	0.788609
.2.24.9.	0.760781
.2.24.10.	0.766532
.2.9.10.	0.766532
.23.3.24.	0.788609
.23.3.9.	0.788609
.23.3.10.	0.788609
.23.24.9.	0.760781
.23.24.10.	0.766532
.23.9.10.	0.766532
.3.24.9.	0.788609
.3.24.10.	0.788609
.3.9.10.	0.788609
.24.9.10.	0.766532
.1.2.23.3.	0.788609
.1.2.23.24.	0.760781
.1.2.23.9.	0.760781
.1.2.23.10.	0.766532
.1.2.3.24.	0.788609
.1.2.3.9.	0.788609
.1.2.3.10.	0.788609

Stage6 (cont.)	
.1.2.24.9.	0.757362
.1.2.24.10.	0.766532
.1.2.9.10.	0.766532
.1.23.3.24.	0.788609
.1.23.3.9.	0.788609
.1.23.3.10.	0.788609
.1.23.24.9.	0.760781
.1.23.24.10.	0.766532
.1.23.9.10.	0.766532
.1.3.24.9.	0.788609
.1.3.24.10.	0.788609
.1.3.9.10.	0.788609
.1.24.9.10.	0.766532
.2.23.3.24.	0.788609
.2.23.3.9.	0.788609
.2.23.3.10.	0.788609
.2.23.24.9.	0.760781
.2.23.24.10.	0.766532
.2.23.9.10.	0.766532
.2.3.24.9.	0.788609
.2.3.24.10.	0.788609
.2.3.9.10.	0.788609
.2.24.9.10.	0.766532
.23.3.24.9.	0.788609
.23.3.24.10.	0.788609
.23.3.9.10.	0.788609
.23.24.9.10.	0.766532
.3.24.9.10.	0.788609
.1.2.23.3.24.	0.788609
.1.2.23.3.9.	0.788609
.1.2.23.3.10.	0.788609
.1.2.23.24.9.	0.760781
.1.2.23.24.10.	0.766532
.1.2.23.9.10.	0.766532
.1.2.3.24.9.	0.788609
.1.2.3.24.10.	0.788609
.1.2.3.9.10.	0.788609
.1.2.24.9.10.	0.766532
.1.23.3.24.9.	0.788609
.1.23.3.24.10.	0.788609
.1.23.3.9.10.	0.788609
.1.23.24.9.10.	0.766532
.1.3.24.9.10.	0.788609
.2.23.3.24.9.	0.788609
.2.23.3.24.10.	0.788609
.2.23.3.9.10.	0.788609
.2.23.24.9.10.	0.766532
.2.3.24.9.10.	0.788609
.23.3.24.9.10.	0.788609

Table C.1: Continued

Stage6 (cont.)	
.1.2.23.3.24.9.	0.788609
.1.2.23.3.24.10.	0.788609
.1.2.23.3.9.10.	0.788609
.1.2.23.24.9.10.	0.766532
.1.2.3.24.9.10.	0.788609
.1.23.3.24.9.10.	0.788609
.2.23.3.24.9.10.	0.788609
.1.2.23.3.24.9.10.	0.788609
27	
.6.	0.781906
28	
.7.	0.796864
29	
.8.	0.781776
30	
.8.	0.750702
Stage7	
31	
.25.	0.771401
.26.	0.72326
.25.26.	0.771401
32	
.25.	0.711409
.26.	0.762831
.25.26.	0.762831
33	
.25.	0.782159
.26.	0.743097
.25.26.	0.782159
24	
.25.	0.786654
.26.	0.704391
.25.26.	0.786654
6	
.27.	0.770514
7	
.28.	0.787078
8	
.29.	0.731814
34	
.30.	0.773383
Stage8	
39	
.31.	0.740581
.32.	0.749632
.33.	0.787568

Stage8 (cont.)	
.24.	0.780087
.31.32.	0.749632
.31.33.	0.787568
.31.24.	0.780087
.32.33.	0.787568
.32.24.	0.780087
.33.24.	0.787568
.31.32.33.	0.787568
.31.32.24.	0.780087
.31.33.24.	0.787568
.32.33.24.	0.787568
.31.32.33.24.	0.787568
40	
.6.	0.710031
38	
.7.	0.777078
29	
.8.	0.756316
30	
.34.	0.734438
Stage9	
35	
.39.	0.791823
.40.	0.782416
.39.40.	0.791823
36	
.40.	0.795041
.38.	0.72813
.29.	0.723274
.30.	0.796939
.40.38.	0.795041
.40.29.	0.795041
.40.30.	0.796939
.38.29.	0.72813
.38.30.	0.796939
.29.30.	0.796939
.40.38.29.	0.795041
.40.38.30.	0.796939
.40.29.30.	0.796939
.38.29.30.	0.796939
.40.38.29.30.	0.796939
Stage10	
23	
.35.	0.706476
24	
.35.	0.799502
6	

Stage10 (cont.)	
.36.	0.771404
7	
.36.	0.75149
8	
.36.	0.757082
TASK2	
Stage2	
1	0.973803
12	0.770361
13	0.765151
37	0.755838
14	0.722836
2	0.923038
3	0.911556
Stage4	
1	
.1.	0.946268
12	
.12.	0.774055
13	
.13.	0.763209
37	
.37.	0.718933
14	
.14.	0.761054
2	
.2.	0.90862
3	
.3.	0.944043
Stage5	
1	
.1.	0.949735
12	
.12.	0.783489
13	
.13.	0.761303
37	
.37.	0.776049
14	
.14.	0.775048
2	
.2.	0.977049
3	
.3.	0.951739

Table C.1: Continued

TASK3	
Stage2	
1	0.961298
12	0.76603
13	0.765911
37	0.718075
14	0.794929
3	0.9289
10	0.935466
Stage4	
1	
.1.	0.969977
12	
.12.	0.79998
13	
.13.	0.788651
37	
.37.	0.778508
14	
.14.	0.757601
3	
.3.	0.956407
10	
.10.	0.981956
Stage5	
1	
.1.	0.957747
12	
.12.	0.785807
13	
.13.	0.72235
37	
.37.	0.702137
14	
.14.	0.72316
3	
.3.	0.95378
10	
.10.	0.941131
TASK4	
Stage2	
15	0.799023
16	0.726778
2	0.928084
9	0.977689

Stage2 (cont.)	
10	0.951449
11	0.978701
Stage4	
15	
.15.	0.755908
16	
.16.	0.738674
2	
.2.	0.973899
9	
.9.	0.998527
10	
.10.	0.990663
11	
.11.	0.935597
Stage5	
15	
.15.	0.793289
16	
.16.	0.708757
2	
.2.	0.905527
9	
.9.	0.99254
10	
.10.	0.925946
11	
.11.	0.901517

Table C.2: Probability of Success of Task 1 with Assets 11 and 2 Assigned after an Enemy Destroys Assets (Graphed in Figure 3.17)

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9180	0.8313	0.7415	0.7137	0.7093	0.3929
Average	0.9180	0.8962	0.8737	0.8511	0.8286	0.8057
Maximum	0.9180	0.9180	0.9180	0.9180	0.9180	0.9180
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.0734	0.0000	0.0000	0.0000	0.0000	0.0000
Average	0.7791	0.7404	0.6739	0.5545	0.3458	0.0000
Maximum	0.9180	0.9169	0.9008	0.8353	0.7547	0.0000

Table C.3: Probability of Success of Task 1 with Assets 11, 2, and 10 Assigned after an Enemy Destroys Assets (Graphed in Figure 3.18)

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9180	0.8314	0.7417	0.7140	0.7130	0.7093
Average	0.9180	0.8963	0.8740	0.8515	0.8292	0.8075
Maximum	0.9180	0.9180	0.9180	0.9180	0.9180	0.9180
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.3929	0.0734	0.0000	0.0000	0.0000	0.0000
Average	0.7857	0.7591	0.7134	0.6168	0.4105	0.0000
Maximum	0.9180	0.9169	0.9008	0.8353	0.7547	0.0000

Table C.4: Probability of Success of Task 1 with Assets 11, 2, 10, and 3 Assigned after an Enemy Destroys Assets (Graphed in Figure 3.19)

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9181	0.8315	0.7420	0.7142	0.7140	0.7130
Average	0.9181	0.8964	0.8741	0.8517	0.8296	0.8080
Maximum	0.9181	0.9181	0.9181	0.9181	0.9181	0.9181
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.7093	0.3929	0.0734	0.0000	0.0000	0.0000
Average	0.7874	0.7665	0.7368	0.6665	0.4754	0.0000
Maximum	0.9180	0.9169	0.9008	0.8353	0.7547	0.0000

Table C.5: Probability of Success of Task 1 with Assets 11, 2, 10, 3, and 1 Assigned after an Enemy Destroys Assets (Graphed in Figure 3.20)

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9182	0.8316	0.7422	0.7144	0.7142	0.7140
Average	0.9182	0.8964	0.8742	0.8518	0.8297	0.8083
Maximum	0.9182	0.9182	0.9182	0.9182	0.9182	0.9181
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.7130	0.7093	0.3929	0.0734	0.0000	0.0000
Average	0.7878	0.7685	0.7479	0.7032	0.5403	0.0000
Maximum	0.9180	0.9169	0.9008	0.8353	0.7547	0.0000

Table C.6: Probability of Success of Task 1 with Assets 11, 2, 10, 3, 1, and 9 Assigned after an Enemy Destroys Assets (Graphed in Figure 3.21)

Probability of Task 1 Success After an Enemy Destroys Assets						
	0 Assets	1 Asset	2 Assets	3 Assets	4 Assets	5 Assets
Minimum	0.9182	0.8316	0.7422	0.7144	0.7142	0.7140
Average	0.9182	0.8965	0.8742	0.8519	0.8298	0.8083
Maximum	0.9182	0.9182	0.9182	0.9182	0.9182	0.9181
	6 Assets	7 Assets	8 Assets	9 Assets	10 Assets	11 Assets
Minimum	0.7138	0.7130	0.7093	0.3929	0.0734	0.0000
Average	0.7879	0.7688	0.7511	0.7268	0.6050	0.0000
Maximum	0.9180	0.9169	0.9008	0.8353	0.7547	0.0000

D. Detailed Description of the Degradation Study

The following is a description of the method used in the degradation study found in section 3.4 so that another analyst can reproduce it with different data. This study used Microsoft's® Excel 2007. Results of the study are located in section 3.4. Throughout this appendix the term "stage" is used in place of "function"

First, the analyst assigned probabilities to the STRATCOM example. The Excel file named "Generate different probabilities for assignables" generates random probabilities for the STRATCOM example. The first sheet is in the same format as the VB code outputs and uses as outlined in section 2.3. The second sheet is a place to input the upper and lower bounds on the uniform random number generator for the assignable and non-assignable assets. When a new random assignment of probabilities is preformed, F9, the combinations of assets automatically finds the maximum and is set. The first two columns from the first sheet were copied and the values were pasted into the first two columns of the first sheet in the Excel file named "Degradation." The macro named "DeleteEndCombo" deletes the rows from the network information that contain "endCombinations." Be sure to delete the first 31 rows by hand to avoid the graphs and summary located to the left of the network information. Column C needs to calculate the probability that the combination of assets listed in Column A is successful and thoughts possible but no listed fail. The macro named "Equation" can be modified to help with the more complex assets. The locations need updated and the correct statements need uncommented for it to work for the particular asset it is used for. Column D multiples Column B by C. Column E sums Column D for the corresponding asset. This is the probability that that asset is successful.

The table located from G1 to J12 is used to assign asset to Task 1 and take assets out of the network. Column G gives the assets that are assignable to Stage 2 Task 1 a sequential number for lookup. Column H is the assets name. The assets with the ability to work on Stage 2 of Tack 1 need to be multiplied by the 0-1 indicator in Column I. These are found in

Columns B for Stage 1 and E for all other Stages for all tasks. The assignable assets' probability of success in Task 1 need multiplied by the 0-1 indicator found in Column J of this table. These probabilities are found in Column B for Stage 1 and Column E for all other stages. In the other tasks the assignable assets' probability of success need multiplied by one minus this 0-1 indicator. Again, these are found in Columns B for Stage 1 and E for the others.

To remove an asset from the probability calculation a macro or user can place the asset numbers corresponding to Column G of the table described in the previous paragraph in Row 3 of the table located from K1 to U3, titled Random Assets. The asset name will appear in Row 2 and the 0-1 indicator in Column G for that asset will be zero.

The probability of success for each of the tasks with the assignable assets assigned according the Column J and the assets in the Random Assets table removed appears in the table located from L5 to O6.

To start the analysis, save the spreadsheet and then set the assignable assets' 0-1 indicators to reflect the desired assignment. Then run the "RunAll" macro. This will run all ways to take the eleven assets out of the network from zero to all eleven. If the new network does not have the same Task 1 Function 2 assets then "remove*n*Assets" macros will need added or removed. These macros iterate through all combinations of removing *n* assets from the network. The summary data and graph on the first worksheet will automatically populate. Once the macro finishes, perform a save as to save the file with a unique name and go back to the original file to run a different assignment. The files titled "Degradation_*x*Task1" contain the data and results for section 3.4 with *x* assets assigned to Task 1.