



**APPLICATION OF A MULTI-OBJECTIVE NETWORK
MODEL TO A COMBAT SIMULATION GAME:
“THE DRIVE ON METZ” CASE STUDY**

THESIS

Timothy D. Frawley, Captain, USAF

AFIT-ENS-14-M-08

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-14-M-08

APPLICATION OF A MULTI-OBJECTIVE NETWORK
MODEL TO A COMBAT SIMULATION GAME:
“THE DRIVE ON METZ” CASE STUDY

THESIS

Presented to the Faculty
Department of Operational Sciences
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

Timothy D. Frawley, B.S.M.E.

Captain, USAF

March 2014

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Abstract

War games are routinely analyzed by the Department of Defense to study the players decision making process. This research develops a multicriteria model that enhances a war game players decision-making capability. The war game consists of a hexagonal-grid map of varying terrain that will be represent as a two-dimensional directed network. The network is obstructed by multiple enemy threats that expose a unit traversing the network to possible attack. The player is faced with the decision of choosing a route to a target node that balances the objectives of following the shortest path and maximizing the probability of success. A weighted arc cost matrix is supplied to Dijkstras shortest path algorithm to find an optimal route. Critical values of the ratio of the objective function weights determine where the optimal path changes. These values are determined on a test scenario for the war game The Drive On Metz.

Table of Contents

	Page
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
I. Introduction	1
1.1 Background	1
1.2 War Gaming	1
1.3 <i>The Drive On Metz</i>	3
1.4 Problem Statement	8
1.4.1 Research Objective	8
1.4.2 Scope	8
1.4.3 Assumptions	8
1.5 Overview of Remaining Chapters	9
II. Literature Review	10
2.1 Game Studies	10
2.2 Vehicle Path Finder	11
2.3 Multicriteria Network Routing	12
2.4 Dijkstra's Algorithm	12
2.5 Analytical Hierarchy Process	13
2.6 Summary	14
III. Methodology	15
3.1 Network Representation	15
3.1.1 Shortest Path Problem	16
3.1.2 Dijkstra's Algorithm	18
3.2 Threat Environment	18
3.2.1 Threat Avoidance	19
3.2.2 Cumulative Probability of Success	20
3.2.3 Threat Avoidance Objective Function	22
3.3 Multicriteria Weighted Sum Model	23

	Page
3.4 Model Implementation	25
3.5 Summary	27
IV. Results and Analysis	28
4.1 Model Behavior	29
4.1.1 Scenario with no threats	29
4.1.2 Scenario with 1 threat	29
4.1.3 Scenario with all threats	31
4.2 Sensitivity Analysis	33
4.3 Objective Ratio	33
4.4 Model Limitations	35
4.5 Summary	37
V. Conclusions and Future Research	38
5.1 Conclusions	38
5.2 Future Research	38
Appendix: MATLAB Code	40
Bibliography	42

List of Figures

Figure	Page
1.1 A portion of <i>The Drive On Metz</i> game map	4
1.2 Unit marker	5
2.1 Dijkstra’s algorithm [1]	13
3.1 Network node identification	16
3.2 Network and distance matrix	17
3.3 Shortest path	19
3.4 Enemy zone of control	19
3.5 Multiple enemy zones of control	20
3.6 Scenario setup form	26
3.7 Model process flow chart	26
4.1 Design space for sensitivity analysis	28
4.2 Shortest path and highest probability path for no threats	30
4.3 Shortest path and highest probability path for one threat	31
4.4 Highest probability path and shortest path for all threats	32
4.5 Contour plot of movement points for $1 \leq \lambda_1 \leq 3$ and $6 \leq \lambda_2 \leq 10$	34
4.6 Contour plot of probability of success for $1 \leq \lambda_1 \leq 3$ and $6 \leq \lambda_2 \leq 10$	34
4.7 Paths for different values of the objective ratio, r	36

List of Tables

Table	Page
1.1 Combat Results Table	6
1.2 Terrain Effects	6
1.3 Awarding of Victory Points	7
2.1 Preference scale for pairwise comparisons [3]	14
3.1 Combat Results Table Probabilities	22
4.1 Movement Points and Probability of Success for Case 1 and Case 2	33
4.2 Objective ratio ranges	35

APPLICATION OF A MULTI-OBJECTIVE NETWORK
MODEL TO A COMBAT SIMULATION GAME:
“THE DRIVE ON METZ” CASE STUDY

I. Introduction

1.1 Background

Many real-world systems can be modeled as a network. The nodes of a network and the arcs between them can represent a wide range of real or conceptual items, such as cities, roads, landmarks, transportation hubs, communication lines, or even human relationships. These connections can have a variety of measurable attributes associated with them, such as distance, cost, weight, or time. One may often encounter the problem of finding a path through this network that is measured or constrained by one or more of the associated attributes of interest. Additionally, there may be obstacles or threats located throughout the network. If an obstacle lies along the desired path, then it may be preferred to reroute through the network to avoid the obstacle. However, this may require the new route to deviate from the preferred path to avoid the obstacle. Furthermore, if there are multiple obstacles, each may pose the entity traveling through the network to a different level of risk. Therefore, the optimal route is one that balances the trade-off between avoiding the high-risk obstacles and following the preferred path.

1.2 War Gaming

In a war game, sometimes referred to as a combat simulation, the battle area is commonly divided into some sort of grid [17]. This structure lends itself to easy modeling as a network, where each grid area can be represented as a node in the network. As a player

plans a strategy and plays the game, many successive decisions are required as to which way is best to maneuver their units. At any time, several courses of action (COAs) may be available for the player to choose from. This decision-making aspect of wargaming is one of its key features.

Perla [15] defines wargaming as “a warfare model or simulation that does not involve the operations of actual forces, and in which the flow of events shapes and is shaped by decisions made by a human player or players.” He further states that war gaming is more focused on the “interplay of human decisions and game events,” rather than just who wins or loses. Ducharme [7] describes the role of wargaming within the armed services as addressing “future concepts and capabilities in the context of Title 10 responsibilities to organize, train, and equip its forces to carry out its roles and functions as a component of the national instrument of power.” The Department of Defense (DoD) routinely studies war games for analytic or training purposes. As opposed to playing for recreation, the DoD desires to gain some useful information or to train their personnel. A war game provides the opportunity to place a player in a decision-making situation and observe their response to a countless number of “what if” scenarios. Perla [15] states that, when wargaming serves the purpose of training, the analysis will focus on “an instructor’s observation and critique of the student’s play.” However, when the purpose of a war game is research, the analysis focuses on “understanding why decisions were made.” He makes a further distinction between wargaming and analysis by saying that wargaming “is a tool for exploring the effects of human interpretation of information rather than those of the actual information (or data) itself. Wargames focus on the decisions players make, how and why they make them, and what effects they have on subsequent events and decisions.”

It is within this decision analysis context that the Air Force Simulation and Analysis Facility (SIMAF) uses the war game *The Drive On Metz* for combat simulation studies. One of their objectives is to conduct studies “with a scientific emphasis on extracting analytical

knowledge from the players” [10]. They have chosen to use a war game approach “to follow the play of the game eliciting decisions, decision criteria, knowledge requirements, thinking patterns, rationale, etc., from all the participants.” When a decision maker, such as a player in a war game, is faced with a decision situation wherein they are faced with multiple objectives to consider, it would be beneficial to have a tool that aids in making a balanced, well-informed decision.

1.3 *The Drive On Metz*

The Drive On Metz is a simple war game that was developed to demonstrate how a war game is designed and played [8]. It attempts to recreate General George Patton’s advance toward the French city of Metz during World War II. With respect to the historical circumstances, the Germans were retreating from the advancing Americans. The Americans sought to cross the Moselle river and capture the town of Metz, while the Germans sought to mount an effective defense that did not expend too much of their resources. *The Drive On Metz* simulates this scenario as a two-player game with one side playing the role of the advancing American forces and the other the defending German forces.

The game is played on a map that represents the vicinity of the city of Metz, covering an area of approximately 1,500 square kilometers. Significant terrain features are interspersed across the map, including cities, towns, fortifications, roads, rivers, clearings, forests, and rough terrain. A hexagonal grid is superimposed on the map with each hexagon covering an area of approximately 4 kilometers across. The hexagonal grid serves the purpose of regulating the location and movement of a player’s forces throughout the game. Figure 1.1 shows a portion of the game map used in *The Drive On Metz*.

The forces for each player consist of a set of markers representing individual units of mostly regimental size. At any point during the game, each hex may be occupied by



Figure 1.1: A portion of *The Drive On Metz* game map

only one unit. Units traverse the map by moving from its current hex to an adjacent hex. Each unit is assigned a fixed *movement allowance* which is used to determine the maximum possible movement of that unit on each turn. Each terrain type requires a different number of movement points to be able to move into that hex. Thus, a unit must have a movement allowance at least as great as the number of movement points required to move into a hex of a particular terrain type. Additionally, all hexes that surround a unit are referred to as that unit's *zone of control*. If a unit enters the zone of control of an enemy unit it must cease its movement.

In addition to being assigned a movement allowance, each unit is also assigned a number representing its *combat strength*. The combat strength is used as a measure to compare the relative strengths of different units and is a key factor in determining the outcome of battle engagements and a player's strategy. The combat strength and movement allowance are depicted on each unit marker as indicated on Figure 1.2.

The game is played over a series of seven turns with each turn consisting of a movement phase and a combat phase for each player. The American player starts the game and each succeeding turn, executing both the movement and combat phases before the

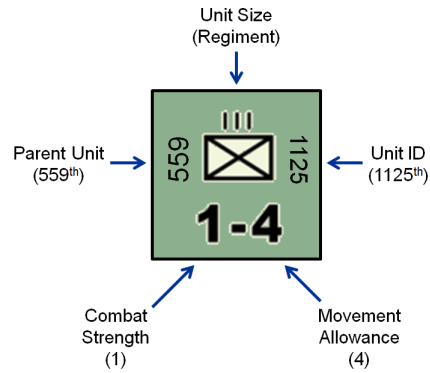


Figure 1.2: Unit marker

German player begins his or her turn. During the movement phase each player may move as many units as they wish, up to the movement allowance of each unit. Once a player has completed all desired movements, they may then engage the other player in combat with as many units that are within the zone of control of the opposing player's units.

The the outcome of a combat engagement is determined by the relative combat strengths of the battling units and the roll of a single die. The combat strength of the defending unit is subtracted from the combat strength of the attacking unit to compute a combat strength differential. A die is then cast and the outcome of the engagement is looked up on the Combat Results Table (CRT). This process is sometimes referred to as the *battle calculus*. The CRT consists of six rows and eight columns in which each row represents the number determined by the die roll and each column represents a combat strength differential. Each entry on the table then indicates the outcome of the engagement by specifying the number of hexes that either the attacker or defender must retreat. Table 1.1 shows the CRT for *The Drive On Metz*.

Additionally, the type of terrain occupied by the defending unit may impact the outcome of a combat engagement. For differing types of terrain, the column referenced

Table 1.1: Combat Results Table

Die Roll	Combat Differential							
	-1	0	+1	+2,+3	+4,+5	+6,+7	+8,+9	+10+
1	—	DR	DR	DR	DR2	DR2	DR2	DR2
2	—	—	DR	DR	DR	DR2	DR2	DR2
3	AR	—	—	DR	DR	DR	DR2	DR2
4	AR	AR	AR	—	DR	DR	DR	DR2
5	AR	AR	AR	AR	—	DR	DR	DR
6	AR	AR	AR	AR	AR	—	DR	DR

“AR” = Attacker Retreats, “DR” = Defender Retreats

on the CRT may be shifted to the left if the terrain occupied by the defender would offer a defensive advantage. The terrain effects on movement and combat are shown Table 1.2.

Table 1.2: Terrain Effects

Terrain Type	Movement Effect	Combat Effect
Road	1 Movement Point	No Shift
Clear	2 Movement Points	No Shift
Rough	3 Movement Points	Shift Left 1 Column
Forest	4 Movement Points	Shift Left 2 Columns
City/Town	Based on terrain	Shift Left 2 Columns
Fortified	Based on terrain	Shift Left 3 Columns

Throughout the game each player may be awarded *victory points* by achieving certain objectives. To win the game, a player must have accumulated more victory points than the

Table 1.3: Awarding of Victory Points

American Side	
5 Points	Per unit east of the Moselle for three complete turns
5 Points	Per unit to exit the east side of the map before the end of the game
5 Points	If an American unit is last to enter or pass through Thionville (hex 0701)
20 Points	If an American unit is the last to enter or pass through Metz (hex 0807)
German Side	
10 Points	Per unit to exit the west edge of the map before the end of the game
7 Points	Per specified unit* to exit the east or south edge of the map on Turn 1
6 Points	Per specified unit* to exit the east or south edge of the map on Turn 2
5 Points	Per specified unit* to exit the east or south edge of the map on Turn 3
4 Points	Per specified unit* to exit the east or south edge of the map on Turn 4
3 Points	Per specified unit* to exit the east or south edge of the map on Turn 5
2 Points	Per specified unit* to exit the east or south edge of the map on Turn 6
1 Points	Per specified unit* to exit the east or south edge of the map on Turn 7

*Applies to the following units: 8 PG, 29 PG, 37 SS, and 38 SS

opponent at the end of the game. The awarding of victory points for each player is shown in Table 1.3. The location of a player's units at specified points in the game determines the awarding of victory points, rather than the outcome of combat engagements. Therefore, battling the enemy is not a direct factor in the outcome of the game, but merely an obstacle to overcome or avoid. The ability to position and maneuver a player's units during play is the crucial factor in determining who ultimately wins the game.

1.4 Problem Statement

The hexagonal structure of *The Drive On Metz* game map lends itself to easy modeling as a connected network. While a player may be able to visually identify a good route on a small map similar to *The Drive On Metz*, it is more difficult for larger or more complex networks. Furthermore, the need to balance competing objectives and the existence of enemy threats that may impede the shortest path make determining an acceptable COA more difficult. Therefore, a tool that considers the objectives and player's preferences in finding an optimal path would greatly enhance the decision-making ability of the player and give them confidence that the chosen COA meets those objectives.

1.4.1 Research Objective.

This research develops a model that can be used as a decision tool for a player in the war game *The Drive On Metz*. The objective of this research is to develop a preference based, multicriteria model that determines an optimal path between any two nodes on the directed network generated from *The Drive On Metz* game map.

1.4.2 Scope.

This model considers the objectives of finding the shortest path and maximizing the probability of successfully completing the path. This research provides the American game player with a tool that improves the ability to select an optimal route for a given static scenario.

1.4.3 Assumptions.

Several assumptions are made to simplify the development of the model:

1. A single American unit is advancing and any number of German units are defending.
2. The model is static.
 - (a) The sequence of turns is not modeled.
 - (b) The threat locations are static.

- (c) The battle calculus (attacks and retreats) is not modeled.
- 3. Known threat locations (*i.e.* perfect knowledge of German unit placement).
- 4. Only one-on-one engagements are considered (*i.e.* no coordinated attacks).

Assumption 1 is necessary to simplify the coding and user interface that is created for the model. Only a one-directional model is needed to demonstrate the effectiveness and validity of the methodology. Assumption 2 is made in keeping within the scope of this research to generate an optimal path for a static point in time. However, any time the situation changes, the model can be run again to determine an updated optimal path. Assumption 3 is a consequence of the standard game rules, whereby both players know the exact location of the other player's units at all times. Finally, Assumption 3 is made to reduce the complexity of the model. This is also a factor in making Assumption 1 because the weaker German units essential require coordinated attacks to defeat the stronger American units.

1.5 Overview of Remaining Chapters

The following is an overview of the remaining chapters. Chapter 2 reviews the literature relevant to developing a multicriteria shortest path model. The methodology for the model is presented in Chapter 3. The results of the model and a demonstration of its functionality for a scenario within *The Drive On Metz* is included in Chapter 4. Finally, Chapter 5 will discuss the conclusions derived from this research and propose some possible areas of future research.

II. Literature Review

A thorough review of the literature is required to develop a multicriteria shortest path model. This chapter summarizes the literature relevant to the development of *The Drive On Metz* model. No sources were found that directly modeled a game as a network. However, there are several instances of a game being the object of an academic study. There are also several sources discussing the application of a network model. Specifically, Dimdal's [6] and Isensee's [12] theses introduce methods for considering multiple objectives in a shortest path network model. Finally, Dijkstra's [5] shortest path algorithm is reviewed.

2.1 Game Studies

Lee's [13] Graduate Research Paper is a recent example that demonstrates the value of using a board game as the object of an academic study. He applied a Markov chain analysis and Monte-Carlo simulation to the board game *RISK* to compare and analyze different game strategies. This project provided an analysis of different strategies that, similar to the objective of this research, could lead to outcomes consistent with a player's preferences. Similarly, Blatt [4], Georgiou [11] and Tan [18] conducted analyses of *RISK* to provide players with a choice of strategies.

Mood [14] describes the value of war games as "the perfect vehicle for studying strategy and tactics... One could try dozens of plans in the time it would take to play dozens of games, and the play of each would test not only its efficacy but its feasibility. The flexibility of a given plan could easily be tested by playing it in the game against a variety of enemy strategies. The sensitivity of a plan to unpredictable factors could be tested by changing those factors over a wide range in the rules of the game." Furthermore, games have beneficial value in training and educational applications, where they can "easily be made to illustrate and clarify complex and subtle relationships."

Mood further presents characteristics of war games that are required for useful analysis. The first is that the game must be easily playable so as to allow it to be played numerous times. This requires a fixed set of rules that allow a player to gain experience that can be carried in later plays. However, in contrast to games with fixed rules, some games may be classified as general-purpose, allowing them to be adapted to differing problem scenarios. In these games, rules and factors may be adjusted so as to examine how those changes impact a player's decisions or how the outcome of the game changes in later plays. In either case, the playing time should be kept short and the number of factors that a player has control over should be limited. This restricts the scope of the game so that it is appropriately sized to the problem at hand—large enough to capture the dynamics of the problem, but no larger. *The Drive on Metz* complies with these characteristics outlined by Mood.

2.2 Vehicle Path Finder

Dimdal [6] applied the A^* algorithm to find an optimal route for a vehicle traveling on a three-dimensional surface. The three-dimensional surface was projected onto a two-dimensional map that was discretized into a regularly-spaced square grid. Each grid cell was assigned attributes characterizing the terrain type and height above sea level. A cell-to-cell approach is used to model the movement across the network. The objective was to find the fastest path between two points, where the vehicle's speed is a function of the terrain type and constrained by the terrain slope, which is determined by the change in elevation between cells.

In finding the unobstructed fastest path, arc costs are first specified as a “basic edge cost”, then a “modifier” is applied to account for enemies or obstacles. The terrain type defines the “basic edge cost”, wherein each terrain type has a different vehicle speed. The “modifiers” then increase or decrease the vehicle speed across that arc to account for any

roads, changes in slope, or obstacles. The modified edge costs were then supplied to the A^* algorithm to find the fastest path.

2.3 Multicriteria Network Routing

Inensee [12] used a multicriteria model to optimize the route of tactical aircraft flying in a radar threat environment. His approach was to model the mission area as a two-level, point-to-point grid network, which was defended by enemy radars. Terrain features, major roads, and cities were modeled as obstacles to be avoided. The two levels represent high and low flight altitudes. The objective was to determine the optimal route considering three criteria: distance traveled, active radar detection, and passive radar detection. Probability of detection was the metric used for radar detection.

Each arc in the network had a set of costs associated with it that correspond to the three objectives. The arc costs for each objective were stored in an individual matrix. A three term composite cost matrix was created by multiplying each of the detection matrices by the distance matrix component-wise, multiplying each by a weight, then summing both of these with the weighted distance matrix. For example, if \mathbf{A} is the distance matrix, \mathbf{B} is the active radar detection matrix, and \mathbf{C} is the passive radar detection matrix, then the composite cost matrix is $\lambda_1\mathbf{A} + \lambda_2\mathbf{AB} + \lambda_3\mathbf{AC}$. The new composite cost matrix is supplied as the arc costs to a modified Dijkstra's algorithm to determine the optimal route.

2.4 Dijkstra's Algorithm

Dijkstra's algorithm [5] is a simple and efficient procedure for finding a shortest path between a source node and all other nodes in a network, assuming non-negative arc lengths [2]. Dijkstra's algorithm is a node-labeling technique that divides the nodes in the network into two sets: those that are temporarily labeled and those that are permanently labeled. The algorithm begins at the source node s and temporarily labels every node in the network with its shortest distance from s . The node with the smallest temporary label

is permanently labeled with its shortest distance, and the temporary labels of all adjacent nodes are updated. This procedure repeats until either all nodes are permanently labeled, or the desired end node is permanently labeled. Throughout this procedure the predecessor of each permanently labeled node is tracked, which allows the shortest path to be backtracked after the procedure terminates. Figure 2.1 shows the algorithm pseudo-code, where S designates the set of permanently labeled nodes, \bar{S} designates the set of temporarily labeled nodes, d_i designates the distance label of node i , $\text{pred}(i)$ designates the predecessor of node i , and c_{ij} designates the cost of each arc (i, j) .

```

algorithm Dijkstra;
begin
   $S := \emptyset$ ;
   $\bar{S} := N$ ;
   $d(i) := \infty$  for each node  $i \in N$ ;
   $d(s) := 0$  and  $\text{pred}(s) := 0$ ;
  while  $|S| < n$  do
    begin
      let  $i \in \bar{S}$  be a node for which  $d(i) = \min\{d(j) : j \in \bar{S}\}$ ;
       $S := S \cup \{i\}$ ;
       $\bar{S} := \bar{S} - \{i\}$ ;
      for each  $(i, j) \in A(i)$  do
        if  $d(j) > d(i) + c_{ij}$  then  $d(j) := d(i) + c_{ij}$  and  $\text{pred}(j) := i$ ;
    end;
  end;

```

Figure 2.1: Dijkstra's algorithm [1]

2.5 Analytical Hierarchy Process

Saaty's [16] Analytical Hierarchy Process (AHP) was reviewed as a potential method of comparing different COAs. The purpose of the AHP is to determine the best COA from a set of potential COAs that is consistent with the decision maker's preferences as rated by the scale shown on Table 2.1. However, the model developed as a result of this research is designed to present a player with a single COA for a given preference with respect to

the objectives. Because alternate COAs can only be generated by changing the player's preferences, it is concluded that use of the AHP is not appropriate for this research.

Table 2.1: Preference scale for pairwise comparisons [3]

Preference Level	Numeric Value
Equally Preferred	1
Equally to Moderately Preferred	2
Moderately Preferred	3
Moderately to Strongly Preferred	4
Strongly Preferred	5
Strongly Preferred to Very Strongly Preferred	6
Very Strongly Preferred	7
Very Strongly Preferred to Extremely Preferred	8
Extremely Preferred	9

2.6 Summary

As the previous game studies described in this chapter demonstrate, war game analysis is a relevant area of study. The methods employed by Dimdal [6] and Isensee [12] of creating weighted arc cost matrices are effective at characterizing a multicriteria network. Furthermore, Dijkstra's algorithm [5] effectively solves the multiobjective shortest path problem. These methods are integrated into *The Drive On Metz* model as described in the subsequent chapters.

III. Methodology

This chapter discusses the methodology of how the multiobjective optimal path model is formulated. To develop this model, it is necessary to describe the network representation of *The Drive On Metz* game map, the representation of threats in the network, the integration and balancing of the primary objectives, and the implementation of the model.

3.1 Network Representation

This study will utilize a directed network, $G = (N, A)$, where N represents the set of nodes and A represents the set of arcs, consisting of all adjacent node pairs (i, j) in the network. Each hex from *The Drive On Metz* game map will represent a node in the network. Each node is connected by up to six arcs joining that node to its adjacent nodes. Because this map contains 9 columns and 11 rows of hexes, the network will contain 99 nodes and 516 arcs. Each node is identified by its column and row indices, n and m , respectively, where $n \in \{01, \dots, 09\}$ and $m \in \{01, \dots, 11\}$. For example, the node in the fourth column and sixth row would be identified as node 0406, which corresponds with its label printed on the game map. Figure 3.1 shows how each hex and its neighboring hexes are identified.

The connecting arc between every adjacent pair of nodes is assigned a cost, c_{ij} . These costs will be further described in a subsequent section of this chapter. These values will be stored in a sparse matrix where each row represents a starting node and each column an ending node. Figure 3.2 shows an example of how the hexagonal grid is extracted from the game map and transformed into the corresponding network and arc cost matrix. For this example, the values in the matrix represent the number of movement points to move between the respective hexes. Consider traveling from node 0102 to node 0202. Observe on the game map that node 0202 is forested terrain, requiring 4 movement points to make

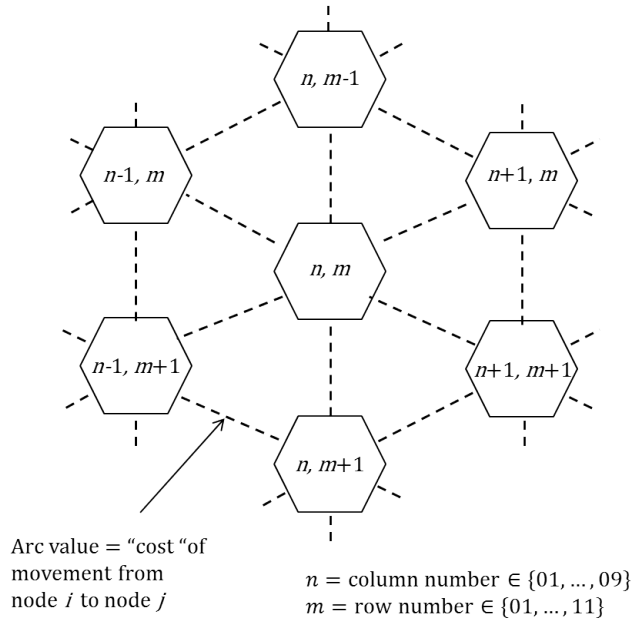


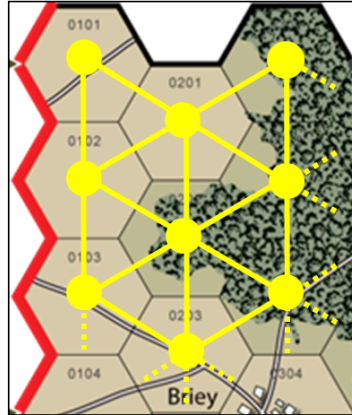
Figure 3.1: Network node identification

this move. Thus, the entry on the distance matrix at (0102, 0202) is 4. The full distance matrix is created in this fashion for all arcs on the game map.

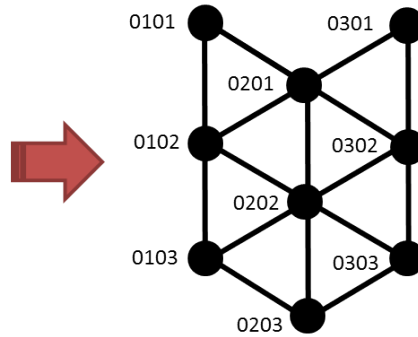
3.1.1 Shortest Path Problem.

With the game map now sufficiently modeled as a directed network, the shortest path between a source node s and a sink node t can be found. The shortest path problem can be solved as a network flow problem in which one unit of flow is introduced into the network [1]. This single unit of flow is allowed to pass through the network from the starting node until it reaches the ending node. The cost of each arc along the path traveled is summed together to determine the total cost of the path. Traveling a particular path that minimizes this cost is the shortest path. The linear programming formulation of the shortest path problem [2] is:

Map and Hexagonal Grid



Network



Distance Matrix

Starting Node	Ending Node								
	0101	0102	0103	0201	0202	0203	0301	0302	0303
0101	0	2		2					
0102	2	0	2	2	4				
0103		2	0		4	1			
0201	2	2		0	4		4	4	
0202		2	2	2	0	2		4	4
0203			1		4	0			4
0301				2			0	4	
0302				2	4		4	0	4
0303					4	2		4	0

Figure 3.2: Network and distance matrix

$$\begin{aligned}
 &\text{Minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 &\text{Subject to} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \text{ or } N \\ -1 & \text{if } i = N \end{cases} \\
 &&& x_{ij} \in \{0, 1\} \quad (i, j) \in A
 \end{aligned}$$

This formulation is for a general network consisting of N nodes and a cost c_{ij} associated with each arc. The variable x_{ij} is a binary variable that indicates whether or not the arc from node i to node j is on the shortest path. This is easily solved using a

linear programming method such as simplex for a small network. However, the formulation can quickly become very large, even for a relatively modest network such as that on *The Drive On Metz*. For the shortest path problem formulated from *The Drive On Metz*, the objective function would contain 516 terms (one indicating whether or not each arc is on the path) and 99 constraints (flow balance at each node). Directly attempting to solve this linear programming formulation would be cumbersome and undesirable. Several effective algorithms exist that can solve a shortest path problem. A common and effective algorithm for solving the shortest path problem is Dijkstra's Algorithm.

3.1.2 Dijkstra's Algorithm.

Dijkstra's algorithm is describe in Section 2.4 and is implemented in *The Drive on Metz* model via a MATLAB function. The MATLAB code is included in Appendix A. The MATLAB function requires the input of the starting node, ending node, and arc cost matrix. As described above, each entry of the cost matrix represents the cost between the two nodes corresponding to that row and column. If the supplied matrix contains values representing the number of movement points between hexes on the game map, the MATLAB code will return the shortest path to travel from the starting node to the ending node. Implementing Dijkstra's algorithm in this fashion will achieve the first objective of minimizing the distance traveled through the network.

3.2 Threat Environment

Simply finding the shortest path may not be sufficient. In many instances, the shortest path may not be feasible or desired if there is some other complicating factor, such as an obstacle along the path. In the context of *The Drive on Metz*, obstacles appear in the form of enemy units. Suppose that for the network shown in Figure 3.3, Dijkstra's algorithm is implemented and the shortest path in terms of movement points from node 0101 to node 0303 is found to be along the path highlighted in blue. Now, suppose an enemy unit is located on node 0103 as shown in Figure 3.4. In order to reach the objective node, a

unit traversing this path will pass into the zone of control of the enemy unit. This would force the advancing unit to stop its movement and either attack the enemy unit or wait to be attacked by the enemy unit. Thus, successfully traversing this path depends on the outcome of the attack. Alternately, another path could be found that completely avoids the enemy zone of control. For example, the path 0101-0201-0302-0303 could be followed to avoid the enemy unit.

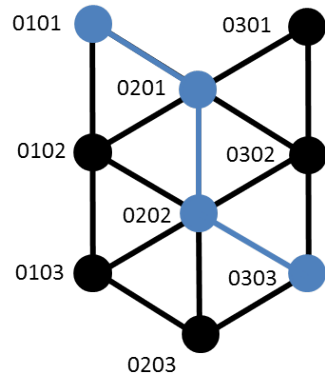


Figure 3.3: Shortest path

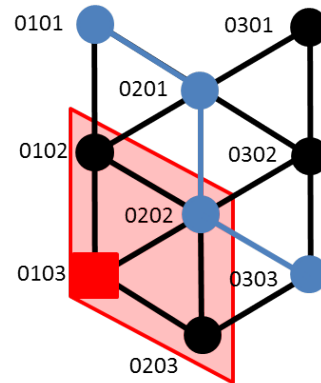


Figure 3.4: Enemy zone of control

3.2.1 Threat Avoidance.

Finding a path that avoids the enemy units will now be considered. A path that completely avoids any nodes that contain an enemy unit is one that maximizes the probability successfully traversing the path. Specifically, the probability of successfully traveling an unimpeded path is 1. However, if there are no feasible paths that can avoid all enemy units, then any path from the starting to ending nodes will have a probability of success that is less than 1. Consider again the network shown in Figure 3.4. If a unit traverses the blue path until the zone of control for the enemy unit located at node 0103 is encountered, then an attack must be successfully resolved in order to continue along that path. For a given combat strength differential and any terrain impacts, the probability of

an attack being successful is determined by the combat results table discussed in Chapter 1 (see Table 1.1). If, for example, the combat strength differential for the units depicted in Figure 3.4 were +4 and there is no terrain impact, then the attack is successful if the defender must retreat, which occurs when the die roll is a 1, 2, 3, or 4. Thus, the probability of success for this attack would be $4/6 \approx 0.67$.

For any node that is overlapped by the zone of control of two or more enemy units, the probability of success for that node is the product of the probabilities associated with all the influential enemy nodes. For example, Figure 3.5 shows that if an additional enemy unit is added at node 0302, then node 0202 is overlapped by two zones of control. If the additional unit has the same individual probability of success as in the previous example, then the total probability of success associated with node 0202 for a single attack would be $0.67^2 \approx 0.44$.

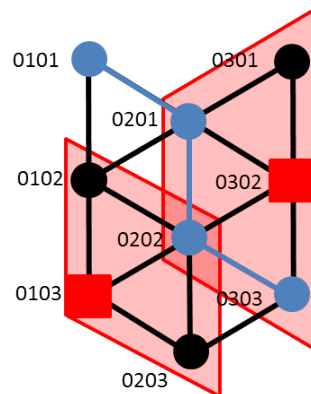


Figure 3.5: Multiple enemy zones of control

3.2.2 Cumulative Probability of Success.

In determining the probability of success against a threat in the examples above, the probability of an attack resulting in a draw is assumed to be the same as a loss. In actuality, there are three possible outcomes of an attack: a win, a loss, or a draw. In the event of

a draw, either player may reattempt the attack on a subsequent turn, again with the same three possible outcomes. Thus, the overall probability of a win is the sum of the probability of a win on the first attempt, plus the probability of a win on any subsequent attempts in the event of a draw. This results in an infinite series that, if taken to the limit, converges to a finite value.

Let PS equal the probability of success and PD equal the probability of a draw. The series expansion representing the overall probability of success would then be:

$$\begin{aligned}
 PS &= PS + PD(PS + PD(PS + PD(PS + \dots))) \\
 &= PS + PD(PS) + PD^2(PS) + PD^3(PS) + \dots + PD^n \\
 &= PS(1 + PD + PD^2 + PD^3 + \dots + PD^{n-1}) + PD^n
 \end{aligned}$$

Approximating this series by computing the probabilities for the first several terms, the probability of a successful attack converges to the values shown in the bottom row of Table 3.1. Recall that “AR” indicates that the attacking unit must retreat (a loss), “DR” indicates that the defending unit must retreat (a loss), and “—” indicates that neither unit retreats (a draw).

Let p_{ij} designate the probability of success along an arc between nodes i and j . Each p_{ij} is fixed for every node in the network when the initial locations of threats are known. For every node j that is not in any enemy unit’s zone of control, the probability of successfully traveling into that node is $p_{ij} = 1$. To find the probability of success for the entire path, take the product of all probabilities along each arc on the path. For the example in Figure 3.5, this would be:

$$\begin{aligned}
 p_{success} &= (p_{0101,0201})(p_{0201,0202})(p_{0202,0303}) \\
 &= (0.67)(0.44)(0.67) \\
 &= 0.20
 \end{aligned}$$

Table 3.1: Combat Results Table Probabilities

Die Roll	Combat Differential							
	-1	0	+1	+2,+3	+4,+5	+6,+7	+8,+9	+10+
1	—	DR	DR	DR	DR2	DR2	DR2	DR2
2	—	—	DR	DR	DR	DR2	DR2	DR2
3	AR	—	—	DR	DR	DR	DR2	DR2
4	AR	AR	AR	—	DR	DR	DR	DR2
5	AR	AR	AR	AR	—	DR	DR	DR
6	AR	AR	AR	AR	AR	—	DR	DR
$P_{success}$	0	0.25	0.4	0.6	0.8	1.0	1.0	1.0

3.2.3 Threat Avoidance Objective Function.

The mathematical objective function that maximizes the probability of success along the path can be written as:

$$\text{Maximize } \prod_{(i,j) \in A} p_{ij}^{x_{ij}}$$

where $x_{ij} \in \{0, 1\}$, $\forall (i, j) \in A$ is the decision variable that indicates if arc (i, j) is on the path.

Taking the natural logarithm, the product can be rewritten as a summation:

$$\text{Maximize } \sum_{(i,j) \in A} (\ln p_{ij}) x_{ij}$$

This objective function can then be written as a minimization by taking its negative.

$$\text{Minimize } \sum_{(i,j) \in A} (-\ln p_{ij}) x_{ij}$$

The only difference between this objective function and the shortest path objective function is that $(-\ln p_{ij})$ is substituted in place of c_{ij} . Thus, this form of the objective function can also be solved via Dijkstra's algorithm as discussed in Section 2.4:

$$\begin{aligned}
& \text{Minimize} && \sum_{(i,j) \in A} (-\ln p_{ij}) x_{ij} \\
& \text{Subject to} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \text{ or } N \\ -1 & \text{if } i = N \end{cases} \\
& && x_{ij} \in \{0, 1\} \quad (i, j) \in A
\end{aligned}$$

Using Dijkstra's algorithm to solve this problem will return a path that has the maximum probability of success. The algorithm will give preference to avoiding threats entirely, but if a zone of control must be traversed, preference will be given to those threats against which the advancing unit has the highest probability of success.

3.3 Multicriteria Weighted Sum Model

Clearly, the two objectives of finding the shortest path and avoiding enemy units may be in direct conflict with one another. Thus, it is desired to integrate both of these competing objectives in a manner that allows for a trade-off of one for the other in accordance with a particular player's preference.

The general multicriteria weighted sum problem formulation [9] is:

$$\begin{aligned}
& \text{Minimize} && \sum_{k=1}^N \lambda_k f_k(x) \\
& \text{Subject to} && x \in X
\end{aligned}$$

for n objectives, where $\lambda_k \geq 0$ is the objective weight and X designates the set of feasible solutions. For this two-objective formulation let:

$$\begin{aligned}
f_1(x) &= d_{ij} x_{ij} \\
f_2(x) &= -(\ln p_{ij}) x_{ij}
\end{aligned}$$

where d_{ij} is the distance measured in movement points. Weighing and summing these two objective functions forms the complete multicriteria objective function:

$$\text{Minimize} \sum_{(i,j) \in A} (\lambda_1 d_{ij} - \lambda_2 \ln p_{ij}) x_{ij}$$

If $\mathbf{D} = [d_{ij}]$ is the distance matrix and $\mathbf{P} = [-\ln p_{ij}]$ is the probability of success matrix for $i, j \in \{0101, \dots, 0911\}$, then the matrix $\mathbf{C} = \lambda_1 \mathbf{D} - \lambda_2 \mathbf{P} = [\lambda_1 d_{ij} - \lambda_2 \ln p_{ij}]$ is the composite arc cost matrix. The value of each term $c_{ij} = (\lambda_1 d_{ij} - \lambda_2 \ln p_{ij})$ in \mathbf{C} is the weighted cost assigned to each arc (i, j) in the network. The final formulation is:

$$\begin{aligned} & \text{Minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\ & \text{Subject to} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \text{ or } N \\ -1 & \text{if } i = N \end{cases} \\ & && x_{ij} \in \{0, 1\} \quad (i, j) \in A \end{aligned}$$

For a specified set of weights λ_1 and λ_2 , this formulation can be solved via Dijkstra's algorithm. The choice of weights will be based on a player's preferred balance of finding the shortest path and avoiding the enemy units.

Despite the fact that the probability term is subtracted from the distance term in the objective function, the terms are really additive because the natural logarithm of a number less than 1 results in a negative value. Thus, subtracting the negative value is really additive. This essentially makes the probability term act like a "modifier" to the distance term. If an enemy unit is present, thus making $\ln p_{ij}$ nonzero, then that term adds to the value of the distance term. This makes that node less unlikely to be on the shortest path. For example, if $d_{ij} = 3$, $p_{ij} = 0.5$, $\lambda_1 = 1$, and $\lambda_2 = 2$, then the weighted arc cost is:

$$\begin{aligned} c_{ij} &= \lambda_1 d_{ij} - \lambda_2 \ln p_{ij} \\ &= (1)(3) - (2) \ln 0.5 \\ &= 3 + 1.4 \\ &= 4.4 \end{aligned}$$

This demonstrates how a node influenced by a threat modifies the cost to traverse an arc that enters that node. The choice of weights will determine the magnitude of that modification, and they can be adjusted in accordance with a player's preferences.

3.4 Model Implementation

The model can be run for a variety of scenarios with the formulation developed in Section 3.3. A basic Decision Support System (DSS) consisting of a simplified version of *The Drive On Metz* was created in Microsoft Excel. The DSS uses a set of Microsoft Visual Basic for Applications (VBA) macros to set up the scenario. The assumptions required to implement the model are listed in Section 1.4.3 and are again summarized here:

1. One American unit is the advancing unit.
2. The German units are the defending units.
3. The German units are stationary.
4. Attacks and combat resolution are not modeled.
5. The probability of success for each arc is independent.

The Excel DSS contains the game map that displays the location of the enemy units and plots the route once it has been determined. All of the pertinent information, including the node and arc lists, distance matrix, threat locations, arc values, combat results table, and unit information are stored within the DSS. The American unit, German units and their locations, the objective weights, and the desired route are updated via a scenario setup form depicted in Figure 3.6.

Figure 3.7 depicts a process chart of how the route is calculated. The arc values are automatically updated once the desired scenario is specified. A VBA function sends the starting node, ending node, and arc values to the MATLAB function which then calculates

The Scenario Setup form is divided into several sections:

- Select American Unit:** US CCA (dropdown), Combat Strength: 7 (input).
- Select Route:** Starting Node: 0101 (dropdown), Ending Node: 0911 (dropdown), Clear button.
- Weight Factors:** Distance Factor: 1.3 (input), Risk Factor: 1 (input), Reset button.
- Select Threats Table:**

German Units	Location	Combat Strength	Differential	Terrain Effect	Probability of Success
GE 1125 Inf	0502	1	6	Shift left 2 columns	0.6
GE 1126 Inf	0403	1	6	Shift left 2 columns	0.6
GE 1010 Inf	0505	2	5	Shift left 2 columns	0.4
GE Utr-Fur	0507	1	6	Shift left 3 columns	0.4
GE Fhnrjr	0509	3	4	Shift left 3 columns	0.25
GE 8 PG	0609	2	5	Shift left 3 columns	0.25
GE 29 PG	0611	2	5	Shift left 1 column	0.6
GE 37 SS	0709	2	5	Shift left 3 columns	0.25
GE 38 SS	0808	3	4	No shift	0.8
GE The Garrison	0807	1	6	Shift left 2 columns	0.6
GE 106 PZ					

Buttons at the bottom: Defaults, Clear All, Show Threats, Get Route, Get Sensitivity, Close.

Figure 3.6: Scenario setup form

the route and outputs the result to a temporary file. Finally, Excel reads the route from the temporary file back into the workbook and plots it onto the map.

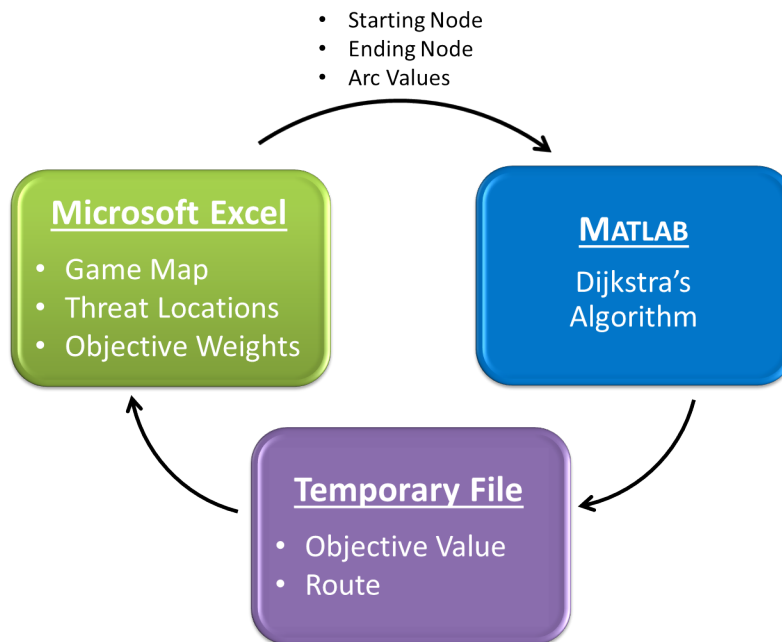


Figure 3.7: Model process flow chart

3.5 Summary

The network representation, incorporation of enemy threats, multiojective weighted arc costs, and DSS are the key components required to implement this model. With these pieces sufficiently describe, the model can be run for a test scenario to determine its behavior.

IV. Results and Analysis

This chapter describes the results and analysis of the model as implemented. The focus is on examining the outcomes of the model for different values of the objective weights in order to describe the overall model behavior. The baseline scenario consists of an American unit attempting to advance from the upper leftmost corner of the map (node 0101) to the lower rightmost (node 0911). The German units are located at the nodes on the map where they would begin during normal game play. This is done by examining a sequence of scenarios to find the shortest path ($\lambda_1 = 1, \lambda_2 = 0$) and the highest probability path ($\lambda_1 = 0, \lambda_2 = 1$) for the cases when there are no threats, one threat, and all threats on the map. Additionally, a sensitivity analysis is done to determine the objective weights at non-endpoint values that result in a change to the optimal path. The objective weights are evaluated across the ranges of $1 \leq \lambda_1 \leq 10$ and $1 \leq \lambda_2 \leq 10$. This results in a square design space consisting of 100 runs as depicted in Figure 4.1

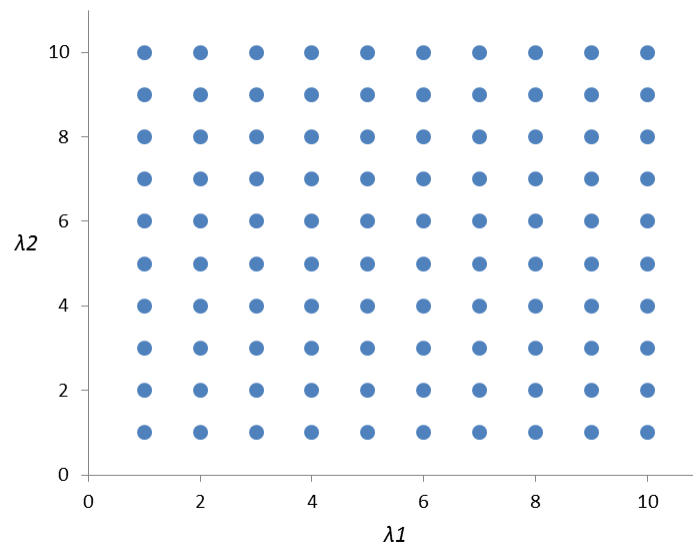


Figure 4.1: Design space for sensitivity analysis

4.1 Model Behavior

This section examines six scenarios for the two cases defined below. The results from these cases are used to validate that the model behaves as expected.

Case 1: $\lambda_1 = 1$ $\lambda_2 = 0$

Case 2: $\lambda_1 = 0$ $\lambda_2 = 1$

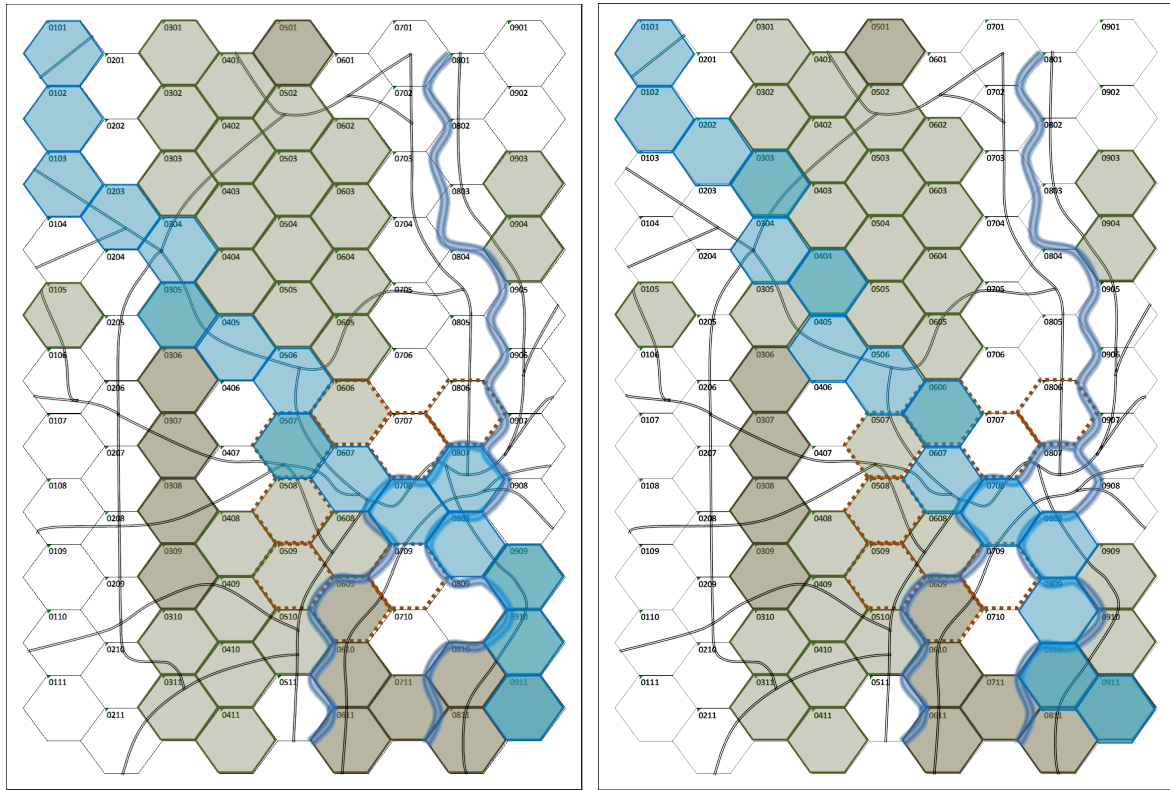
4.1.1 Scenario with no threats.

Figure 4.2a shows the shortest path for the Case 1 baseline scenario on the game map strictly in terms of movement points. As the figure shows, this path follows a roadway since that is the least cost type of terrain to travel. The number of movement points required to complete this route is 19.

Figure 4.2b shows the path with the highest probability of success for Case 2. Because there are no threats impeding this path, the probability of success is 100%. This path is similar to the shortest path, however, the algorithm is no longer constrained by the terrain in finding the path. Rather, it traverses the fewest number of arcs that still result in the highest probability. This results in a route that does not follow a road and instead passes through undesirable terrain, including several river crossings where there is no road, and thus requires 58 movement points to complete. This is significantly higher than the shortest path.

4.1.2 Scenario with 1 threat.

Again consider Case 1 and suppose a single threat is placed on the map such that it lies on the shortest path as shown in Figure 4.3a. The threat shown is the German *Unterfuhrer* regiment, which begins the game at node 0507 and has a combat strength of 1. Assuming the American unit has the highest possible combat strength of 7, the resulting combat differential is +6. Furthermore, because the German unit lies on a node that is fortified, a shift of 3 columns to the left on the combat results table is permitted. Thus, by applying the probability of success method describe in Section 3.2.2, the probability of



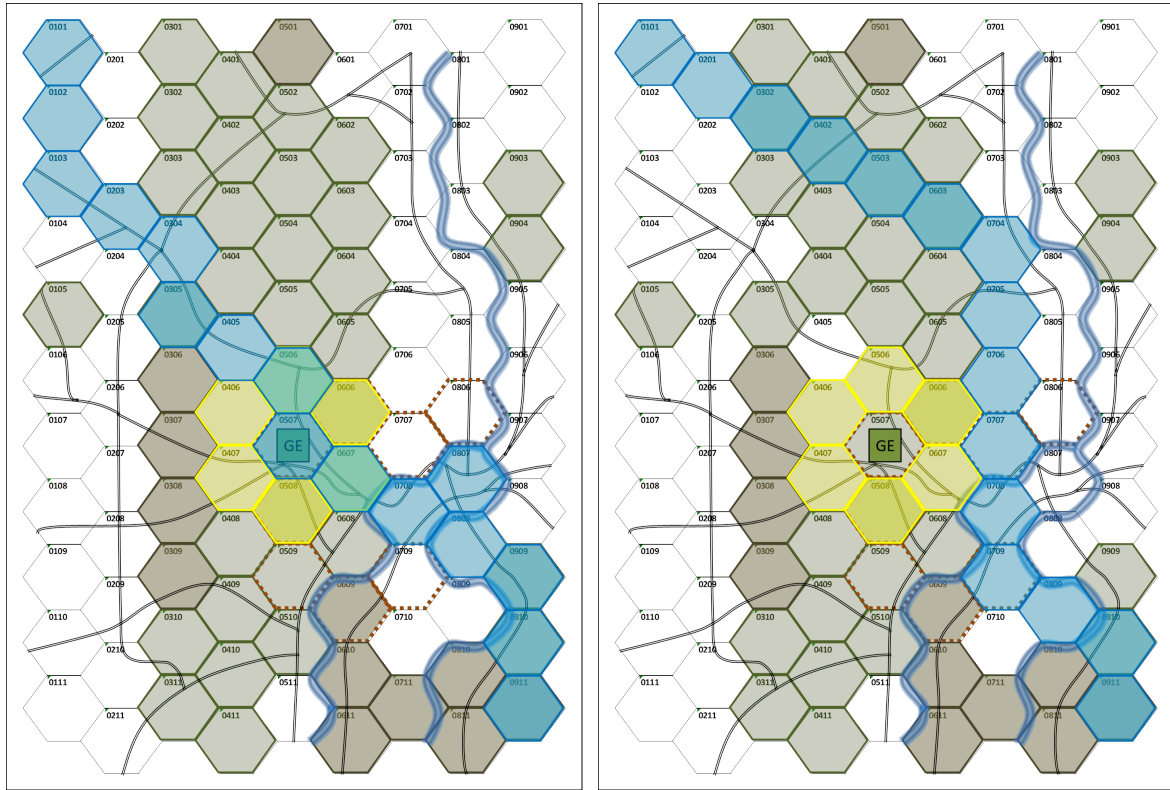
(a) Shortest Path

(b) Highest Probability

Figure 4.2: Shortest path and highest probability path for no threats

success against this unit is 0.4, or 40%. However, because the model assumes stationary threats and this route passes through three nodes in the German unit's zone of control, the overall probability of success computed by the model for this path is $0.4^3 = 0.064$, or 6.4%.

Figure 4.3b shows how the path changes for Case 2 when the same threat is in place. As in the case with no threat, the algorithm finds a path that has an overall probability of success equal to 100%, but again does not consider the terrain being traveled through. Thus, the number of movement points for this path is 43.



(a) Shortest Path

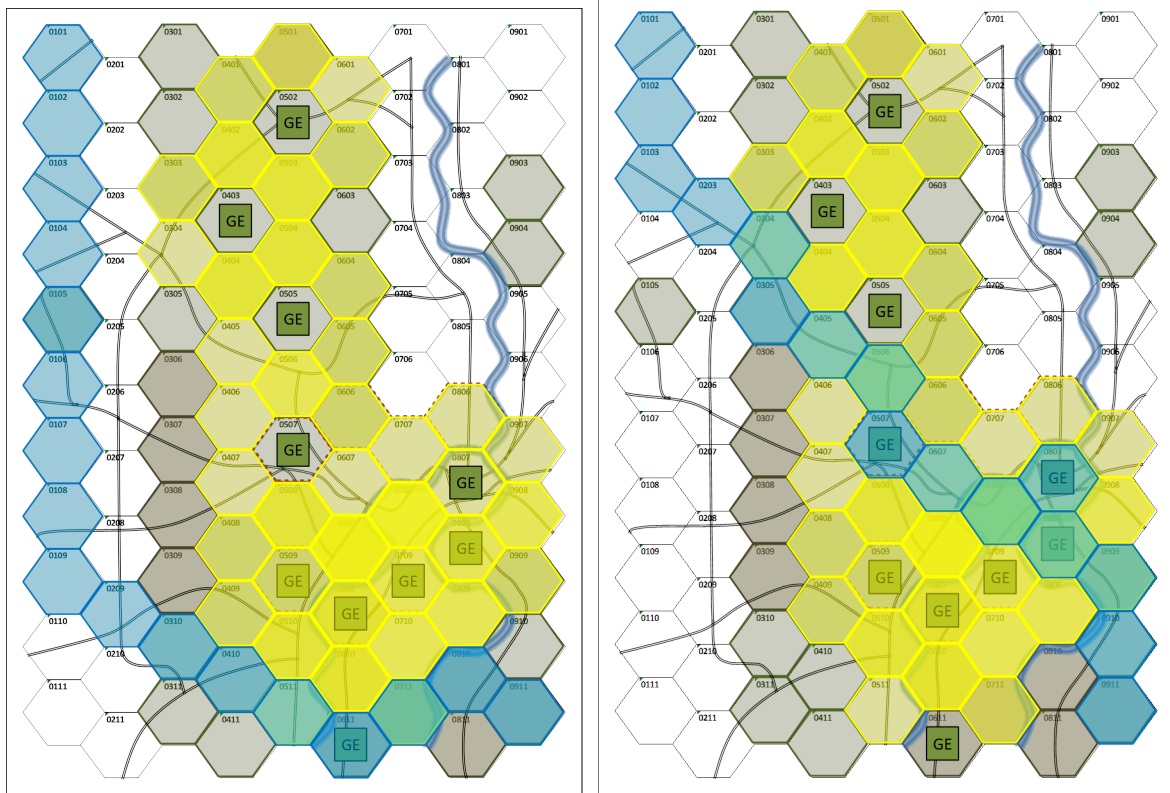
(b) Highest Probability

Figure 4.3: Shortest path and highest probability path for one threat

4.1.3 Scenario with all threats.

Consider now the scenario where all German units are on the map at their initial positions. Figure 4.4b shows the shortest path for Case 1 with this threat layout. As before, this path requires 19 movement points to traverse, but now there are multiple threats located on the path. The zone of control for six German units must be passed through, thus the overall probability of success for this path is 3.4×10^{-5} , or essentially 0%.

Figure 4.4a shows how the path changes for Case 2 with the same threat locations. In this case, only one German zone of control is passed through, and the probability increases to 0.216, or 21.6%. However, the number of movement points drastically increases to 55.



(a) Highest Probability Path

(b) Shortest Path

Figure 4.4: Highest probability path and shortest path for all threats

Table 4.1 summarizes the results for both Case 1 and Case 2 with each threat layout. These results show that the model behaves as expected. In the extreme case where finding the shortest path is the only objective being considered, the model finds the same shortest path regardless of the presence of enemy threats. Conversely, in the extreme case where finding the path with the highest probability of success is the only objective being considered, the model finds a route that avoids all threats as much as possible.

Table 4.1: Movement Points and Probability of Success for Case 1 and Case 2

Threat Scenario	No Threats		One Threat		All Threats	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
Movement Points	19	58	19	43	19	55
Probability of Success	100%	100%	6.4%	100%	0.0%	21.6%

4.2 Sensitivity Analysis

To explore the effects of the objective weights λ_1 and λ_2 , the model was iteratively run with each weight varying from $1 \leq \lambda_1 \leq 10$ and $1 \leq \lambda_2 \leq 10$ in step sizes of 1. Because the objective function contains mixed units of movement points and probabilities, the value of the objective function itself is not particularly meaningful. Rather, a player would be more interested in the number of movement points and the probability of success to traverse a path. Figure 4.5 shows a contour plot that depicts the decision space of the movement points for ranges of λ_1 and λ_2 that result in the most change. This plot shows that as the shortest path weight λ_1 increases or the probability of success weight λ_2 decreases, the resulting path length decreases. Conversely, Figure 4.6 shows that the trend is reversed for the probability of success in the same range of objective weight values. As the shortest path weight λ_1 decreases or the probability of success weight λ_2 increases, the probability of successfully traversing the path increases.

4.3 Objective Ratio

In order to better compare the results of the sensitivity analysis, it is useful to express the results in terms of the objective ratio, here defined as the *preference parameter*, r :

$$r = \frac{\lambda_1}{\lambda_2}$$

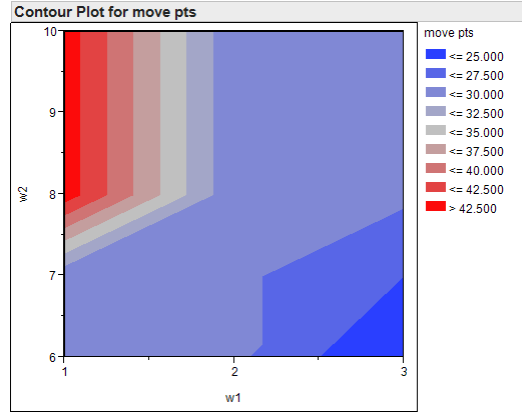


Figure 4.5: Contour plot of movement points for $1 \leq \lambda_1 \leq 3$ and $6 \leq \lambda_2 \leq 10$

Note: “w1” = λ_1 and “w2” = λ_2 in this figure

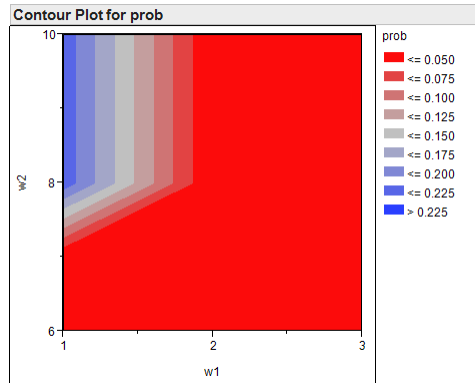


Figure 4.6: Contour plot of probability of success for $1 \leq \lambda_1 \leq 3$ and $6 \leq \lambda_2 \leq 10$

Note: “w1” = λ_1 and “w2” = λ_2 in this figure

Higher values of r result in paths that favor the shortest path, whereas lower values r will favor a path with the highest probability of success. The critical values of r where the path changed during execution of the sensitivity analysis for the baseline scenario are show in Table 4.2. This table shows that r values above 1.6 do not result in any changes in the

shortest path. However, r values below 1.6 progressively trade off the shortest path for a path with a higher probability of success. It is important to note that these critical values are applicable only to this particular scenario. Differing threat locations or an American unit with a different combat strength may result in different critical values.

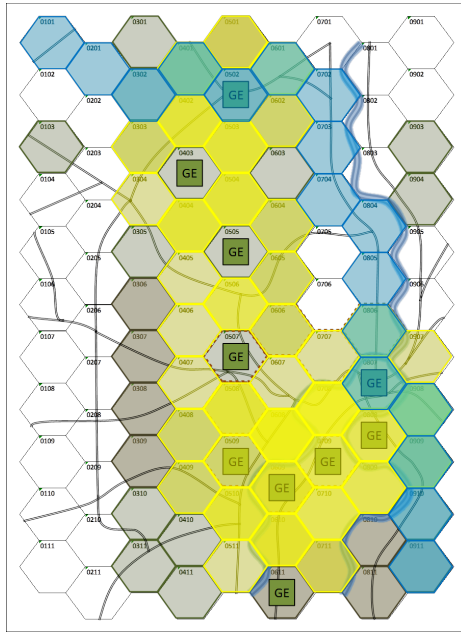
Table 4.2: Objective ratio ranges

COA	Range for r	Movement Points	Probability of Success
1	[0, 0.13)	44	21.6%
2	[0.13, 0.40)	28	2.4%
3	[0.40, 0.46)	25	0.7%
4	[0.46, 1.17)	22	0.2%
5	[1.17, 1.60)	20	0.02%
6	[1.60, ∞)	19	0.0%

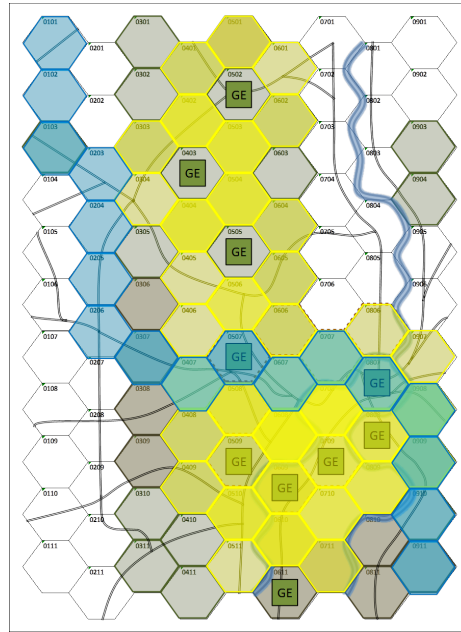
Figures 4.4a and 4.4b are examples of paths for r values in the first and last ranges of Table 4.2, respectively. Figures 4.7a to 4.7d show examples of the paths for increasing values of r in each of the other four respective ranges.

4.4 Model Limitations

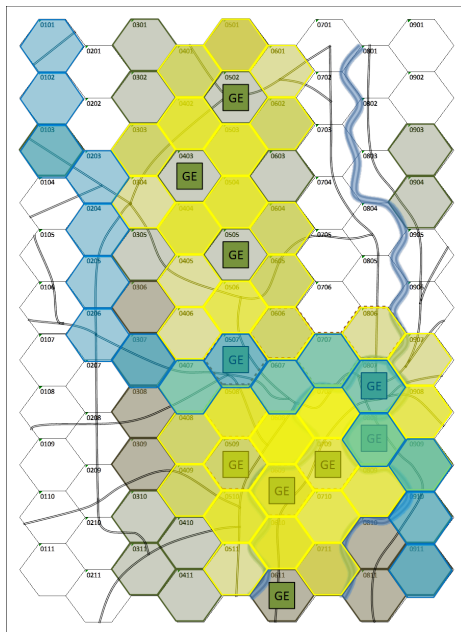
Although this research has developed a model that effectively balances a player's preferences in finding an optimal route, there are some innate limitations. One such limitation is its speed. While the model effectively computes an optimal path, it takes approximately ten seconds to compute the route. While this is sufficient for an academic study of the optimal route, it may be cumbersome for a player to use during the course of the game. Additionally, if this methodology were expanded to larger network applications, the computation speed will also increase.



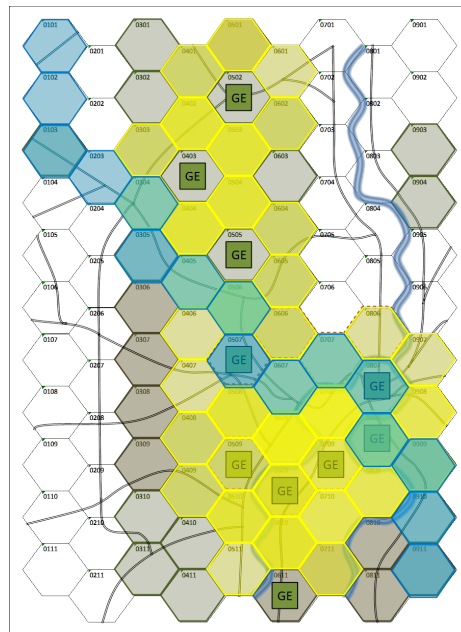
(a) $r = 0.25$



(b) $r = 0.45$



(c) $r = 1.0$



(d) $r = 1.5$

Figure 4.7: Paths for different values of the objective ratio, r

Another limitation of this methodology is that there is no mechanism to determine alternate optimal routes. This is inherent to the use of Dijkstra's algorithm. There is no way of knowing if an equally optimal route is available if there are extraneous factors that would make the route determined by the model undesirable. Thus, the player would be forced to compromise on their preference in order to find another route.

Finally, the assumptions describe in Section 1.4.3 outline several limitations. As describe in that section, many were made to reduce the complexity of the model. However, this results in the model having reduced fidelity in recreating the actual game play experience of *The Drive On Metz*.

4.5 Summary

The results of this model show that it behaves as expected. Changes to the preference parameter, r , accurately produce a corresponding change in the model output, where higher values of r generate a shortest path route and lower values of r generate a higher probability of success route. Therefore, the preference parameter is an appropriate characterization of a game player's preference towards finding the shortest path or maximizing the probability of successfully completing the path.

V. Conclusions and Future Research

5.1 Conclusions

This research applies a network model to the war game *The Drive On Metz*. The result is a multicriteria model that can be a useful tool for a game player to balance the objectives of finding the shortest path and finding the path that avoids threats with the highest probability of success. The model behavior is consistent with the expected outcomes for a given set of inputs. The preference parameter r effectively quantifies the relative importance of the two objectives and characterizes the player's risk aversion. For a given threat layout an specified value of r , the model will compute an optimal route. By considering different values of r , the game player is then provided a range of efficient paths from which to chose. This successfully demonstrates how the model accommodates varying levels of risk aversion and consequently produces appropriately adjusted outcomes.

5.2 Future Research

The natural progression from this research would be to address the limitations and assumptions. The computation speed of the model could be increased by reworking the VBA code to be more efficient or using a more advanced programming language or technique. Efficiency can also be gained by examining the shortest path algorithm. Dijkstra's algorithm could be enhanced to improve upon its shortcomings and improve its data structure, or another algorithm that is more efficient could be used. Similarly, incorporating an algorithm or technique to search for alternate optimal routes could provide the game player with multiple COAs for a given preference parameter.

To improve the fidelity of the game play experience, more functionality of the game could be incorporated. This could include multiple advancing units for coordinated attacks, modeling the sequence of turns for both players, or simulating the attacks and combat

resolution. Because this model is based on a static scenario, a logical progression could be to incorporate a dynamic threat environment. A dynamic programming approach could consider the outcome of an attack against a threat encountered along the path, then recompute the optimal path from that point. This may provide a better picture of whether or not a particular route is preferable.

A pre-assessment of a player's risk aversion may be useful in estimating the preference parameter for that player. That value could then be input into the model and held constant throughout the game to determine if it is consistently accurate.

It may also be of interest to consider the possibility of incorporating nodes of interest in the network that a route might be drawn to, rather than only considering a network with nodes that contain threats to be avoided. This could be done by reversing the penalty approach and instead providing an incentive towards particular arcs or nodes of interest.

Finally, it may be of interest to consider a non-perfect information format of the game. This could be done by masking the unknown threat locations and applying a probability to the likelihood of a threat at a particular location. This probability would be a measure of confidence in the reliability or quality of the imperfect information.

Appendix: MATLAB Code

Listing A.1: MATLAB code

```
1 function [] = metz_route(RouteStart, RouteEnd)
2
3 clc
4
5 % Excel data file variables used in the functions below.
6 filename1 = 'Metz_macro.xlsxm';
7 filename2 = 'temp.xlsx';
8 sheet1 = 'Arcs';
9 sheet2 = 'Nodes';
10 sheet3 = 'Route';
11 sheet4 = 'Threats';
12 sheet5 = 'Laydown';
13 sheet6 = 'Distance';
14
15 % 'xlsread' is a built in MATLAB function used to read the
16 % distance matrix values as vectors from the Excel file.
17 Distance = xlsread(filename1,sheet5,'D:D');
18 StartNodes = xlsread(filename1,sheet5,'E:E');
19 EndNodes = xlsread(filename1,sheet5,'F:F');
20
21 % 'sparse' is a built in MATLAB function that generates a
22 % sparse matrix from three vectors defining the rows,
23 % columns, and matrix values, respectively. This matrix
24 % represents the directed graph of the network.
```

```

25 DG = sparse(StartNodes,EndNodes,Distance);
26
27 % 'graphshortestpath' is a built in MATLAB function that takes
28 % sparse network, starting node, and ending node as inputs and
29 % finds the shortest path. 'dist' is the length of the shortest
30 % path, 'path' is a vector of nodes representing the path, and
31 % 'pred' is a vector of nodes representing the predecessor nodes.
32 % Dijkstra's algorithm is this functions default method for
33 % finding the shortest path.
34 [dist, path, pred] = graphshortestpath(DG,RouteStart,RouteEnd);
35
36 % 'xlswrite' is a built in MATLAB function that exports the
37 % results to a temporary Excel file
38 xlswrite(filename2,transpose(path),sheet3,'A1');
39 xlswrite(filename2,dist,sheet6,'A1');
40 end

```

Bibliography

- [1] Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Inc., 1993.
- [2] Bazaraa, Mokhtar S., John J. Jarvis, and Hanif D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Inc., 4th edition, 2010.
- [3] Bernard W. Taylor, Sr. *Introduction to Management Science*. Pearson Education, Inc., 10th edition, 2010.
- [4] Blatt, Sharon. "RISKy business: An in-depth look at the gameRISK". *Undergraduate Math Journal*, 3, 2002.
- [5] Dijkstra, Edsger W. "A note on two problems in connexion with graphs". *Numerische mathematik*, 1(1):269–271, 1959.
- [6] Dimdal, Markus. *An Optimal Pathfinder for Vehicles in Real-World Terrain Maps*. Master's thesis, The Royal Institute of Science, School of Engineering Physics, Stockholm, Sweden, 2003.
- [7] Ducharme, Douglas. "Approaches to Title 10 Gaming", January 2014. URL <http://www.usnwc.edu/Research---Gaming/War-Gaming/Documents/Publications/Articles.aspx>.
- [8] Dunnigan, James F. *The Complete Wargames Handbook: How to play, design, and find them*. William Morrow & Co., Inc., revised edition, 1992.
- [9] Ehrgott, Matthias. *Multicriteria Optimization*. Springer, 2nd edition, 2005.
- [10] Garrambone, Michael. "Intel Agent System Project". Microsoft PowerPoint Presentation, July 2013. Presented at InfoSciTex, Beavercreek, OH.
- [11] Georgiou, Harris. "RISK Board Game-Battle Outcome Analysis". 2004.
- [12] Isensee, Ernst K. *Multicriteria Network Routing of Tactical Aircraft in a Threat Radar Environment*. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 1991.
- [13] Lee, Jordan D. *The Comparison of Strategies used in the game of RISK via Markovian Analysis and Monte-Carlo Simulation*. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH. Graduate Research Project.
- [14] Mood, Alexander M. *War Gaming As A Technique of Analysis*. Technical report, 1954.

- [15] Perla, Peter P. *The Art of Wargaming: A guide for Professionals and Hobbyists*. United States Naval Institute Press, 1990.
- [16] Saaty, Thomas L. “How to Make a Decision: The Analytic Hierarchy Process”. *Interfaces*, 24(6):19–43, Nov-Dec 1994.
- [17] Sabin, Philip. *Simulating War: Studying Conflict through Simulation*. Continuum International Publishing Group, 2012.
- [18] Tan, Baris. “Markov chains and the RISK board game”. *Mathematics Magazine*, 70(5):349–357, 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 27-03-2014		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Oct 2012–Mar 2014	
4. TITLE AND SUBTITLE APPLICATION OF A MULTI-OBJECTIVE NETWORK MODEL TO A COMBAT SIMULATION GAME: “THE DRIVE ON METZ” CASE STUDY				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
6. AUTHOR(S) Frawley, Timothy D., Captain, USAF				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENS-14-M-08	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Simulation and Analysis Facility Wright-Patterson AFB, OH 45433 Randyll L. Levine, GS-15, DAF Chief, Simulation and Analysis Div (AFLCMC/XZS) COMM (937) 938-3777 [DSN 798-3777]				10. SPONSOR/MONITOR'S ACRONYM(S) AFLCMC/XZS	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT War games are routinely analyzed by the Department of Defense to study the players decision making process. This research develops a multicriteria model that enhances a war game players decision-making capability. The war game consists of a hexagonal-grid map of varying terrain that will be represent as a two-dimensional directed network. The network is obstructed by multiple enemy threats that expose a unit traversing the network to possible attack. The player is faced with the decision of choosing a route to a target node that balances the objectives of following the shortest path and maximizing the probability of success. A weighted arc cost matrix is supplied to Dijkstras shortest path algorithm to find an optimal route. Critical values of the ratio of the objective function weights determine where the optimal path changes. These values are determined on a test scenario for the war game The Drive On Metz.					
15. SUBJECT TERMS War game, combat simulation, network, multicriteria, multiobjective, shortest path					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. James W. Chrissis (ENS)
U	U	U	UU	53	19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4606 james.chrissis@afit.edu