



**Applying the Multiple Multidimensional  
Knapsack Assignment Problem to a Cargo  
Allocation and Transportation Problem with  
Stochastic Demand**

THESIS

Jocelin S. Maus, Captain, USAF  
AFIT-ENS-MS-19-M-137

**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 document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Army, the United States 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-MS-19-M-137

APPLYING THE MULTIPLE MULTIDIMENSIONAL KNAPSACK  
ASSIGNMENT PROBLEM TO A CARGO ALLOCATION AND  
TRANSPORTATION PROBLEM WITH STOCHASTIC DEMAND

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

Jocelin S. Maus, B.S.

Captain, USAF

March 2019

DISTRIBUTION STATEMENT A  
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENS-MS-19-M-137

APPLYING THE MULTIPLE MULTIDIMENSIONAL KNAPSACK  
ASSIGNMENT PROBLEM TO A CARGO ALLOCATION AND  
TRANSPORTATION PROBLEM WITH STOCHASTIC DEMAND

THESIS

Jocelin S. Maus, B.S.  
Captain, USAF

Committee Membership:

Maj Thomas P. Talafuse, Ph.D.  
Chair

Lt Col Jeremy D. Jordan, Ph.D.  
Co-Chair

## **Abstract**

The US military relies on airlift to not only deploy and sustain U.S. armed forces anywhere in the world but also to rapidly mobilize humanitarian efforts and supplies. Operations already impacted by the limited capacity of aircraft also fall prey to dynamic requirements and differing priorities of multiple global locations. A growing concern for the modern military budget is how to provide airlift functions expediently and economically while mitigating the costs of shortfalls and overages. Utilizing fiscal year 2017-2018 cargo data published by the 618th Air Operations Center and modeling this problem as a multiple multidimensional knapsack assignment problem (MMKAP), this work investigates how categorical assumptions about demand affect aircraft allocation and assesses the economic penalties associated with shorting or exceeding demand in the event of mis-estimation given a stochastic demand.

This work starts with the general formulation of a new variant of the MMKAP and applies the MMKAP to a notional military airlift example with two supply, two demand nodes, two item types, and three aircraft types. After a deterministic solution is found, the effects of a stochastic demand are explored using different cost models and random draws from distribution functions based on reported cargo shipment data. This research concludes that there are levels at which demand expectations can be set to mitigate economic penalties given a fixed cost penalty and a variable cost penalty.

## Acknowledgements

I would like to first thank my advisors, Maj Talafuse and Lt Col Jordan, for their guidance and flexibility through the variations of this research. Without them, there would not be a product. Secondly, I would like to thank Dr. Lunday for the time and effort he dedicated to helping with model formulation, GAMS troubleshooting, and mental support. If I ever enter the world of academia as an educator I would aspire to be like him. Also, I would like to thank Dr. Hill for being there at the crucial moment when I needed an objective set of eyes. His intelligence, experience, and caring saved me from a world of embarrassment. Lastly, I would like to thank my husband for his resilience through this undertaking. He is my constant.

Jocelin S. Maus

# Table of Contents

	Page
Abstract .....	iv
Acknowledgements .....	v
List of Figures .....	viii
List of Tables .....	x
I. Introduction .....	1
1.1 Background .....	1
1.2 Motivation .....	2
1.3 Research Approach and Objectives .....	2
1.4 Organization .....	3
II. Literature Review .....	4
2.1 Overview .....	4
2.2 Multiple Knapsack .....	5
2.3 Multidimensional Knapsack .....	6
2.4 Assignment Problem .....	7
2.5 Multiple Knapsack Assignment Problem .....	8
2.6 Branch-and-Bound Method .....	9
2.7 Branch-And-Reduce Optimization Navigator (BARON) .....	9
2.8 Aircraft Routing and Resources .....	11
III. Methodology .....	12
3.1 Introduction .....	12
3.2 General Mathematical Model .....	12
3.3 Complexity .....	14
3.4 Fuel Estimation .....	15
3.5 Distance Estimation .....	18
3.6 Applied Mathematical Model .....	19
3.7 Assumptions .....	22
3.8 General Algebraic Modeling System Implementation .....	23
3.9 Solution Pre-Processing .....	24
3.10 Solution Processing .....	27
3.11 Demand .....	28

	Page
IV. Analysis .....	29
4.1 Introduction .....	29
4.2 Pre-Analysis Processing .....	29
4.3 Actualizing Demand .....	31
4.4 Aircraft Allocation .....	32
4.5 Fixed Cost Models .....	36
4.6 Variable Cost Model .....	42
V. Conclusions and Future Research .....	44
5.1 Conclusion .....	44
5.2 Future Research .....	45
Appendix A. Visual Basic for Applications Code .....	46
1.1 Fuel .....	46
1.2 Distance .....	48
1.3 Pre-Processing .....	54
1.4 Post-Processing .....	57
Appendix B. General Algebraic Modeling System Code.....	60
Bibliography .....	63

## List of Figures

Figure		Page
1	BARON Algorithm [1] .....	10
2	Aircraft Selection Form .....	25
3	Graph Comparison of GAMS and Post-Processed Final Cost vs. Number of Pallets .....	31
4	Graph Cost per Pallet vs. Number Pallets: No Priority Cost .....	31
5	Cost Per Pallet vs. Number of Pallets: Comparison of MDS Only and Mixed Profiles .....	33
6	Graph Final Cost vs. Number of Pallets, Separate MDS Solution .....	33
7	Graph Cost Per Pallet vs. Number of Pallets, Separate MDS Solution .....	34
8	Number of Aircraft vs. Number of Pallets: Mixed Profile .....	34
9	Number of Aircraft vs. Number of Pallets: Separate Profiles .....	35
10	Histogram of Super/999/1 Cargo .....	35
11	Histogram of Priority 2/3 Cargo .....	36
12	Cost Per Pallet vs. Fixed Penalty Cost for Overage .....	37
13	Percentile vs. Fixed Penalty Cost for Overage .....	38
14	Cost Per Pallet vs. Fixed Penalty Cost for Shortage .....	39
15	Percentile vs. Fixed Penalty Cost for Shortage .....	39
16	Cost Per Pallet vs. Fixed Penalty Cost .....	40
17	Percentile vs. Fixed Penalty Cost .....	40
18	Percentile vs. Fixed Penalty Cost: Super/999/1 Cargo .....	41
19	Percentile vs. Fixed Penalty Cost: Priority 2/3 Cargo .....	41

Figure		Page
20	Cost Per Pallet vs. Percent of Pallet Cost for Overage Penalty.....	43
21	Percentile vs. Percent of Pallet Cost for Overage Penalty.....	43

## List of Tables

Table		Page
1	Aircraft Specific Payload Movement Assumptions [2] .....	15
2	Climb $\omega_{fc}$ Regression Terms [2] .....	16
3	Descent $\omega_{fd}$ Regression Terms [2] .....	16
4	Cruise $\omega_{ff}$ Regression Terms [2] .....	17
5	Quantiles for Expected Demand .....	30
6	Data Parameters per Priority Type .....	36

APPLYING THE MULTIPLE MULTIDIMENSIONAL KNAPSACK  
ASSIGNMENT PROBLEM TO A CARGO ALLOCATION AND  
TRANSPORTATION PROBLEM WITH STOCHASTIC DEMAND

## I. Introduction

### 1.1 Background

The US military relies on airlift to not only deploy and sustain U.S. armed forces anywhere in the world but also to rapidly mobilize humanitarian efforts and supplies. A growing concern for the modern military budget is how to provide airlift functions expediently with economical awareness. Research on the subject of cargo transportation via aircraft has existed as long as the advent of military airlift itself. With it, a myriad of paths to improve cargo transportation have emerged. Historically, to improve cargo throughput, aircraft designers emphasized larger and faster planes, however, recently these efforts have plateaued. New planes are marginally better than old in the realm of size and speed. Recent considerations of the improvement of air transportation have delved into fuel efficiency. Reiman proposed improving fuel efficiency and cargo throughput through alternative routing methods [2]. Boone presented a methodology that incorporates ensemble, versus deterministic, numerical weather prediction models into route planning, thus reducing the amount of excess fuel burned by poor forecasts and providing a range of potential values which aid in flight planning [3].

## 1.2 Motivation

Optimization has been widely used in the civilian sector to save costs, promote efficiency, and reduce waste. While government entities should endeavor to optimize for the same reasons, they also utilize optimization to conserve manpower, improve lethality, and save lives. Supplying troops and humanitarian aid is a military transportation problem. Maywald *et al.* created the Aircraft Selection Model (ASM) that utilizes the routing methods and fuel regression equations of Reiman to select aircraft to efficiently or economically transport given cargo from one source to one destination [4][2]. However, operations already impacted by the limited capacity of aircraft also fall prey to dynamic requirements and differing priorities of multiple global locations. The question is how to assign and utilize varying cargo aircraft types to these demands to minimize cost and maximize priority fulfillment.

## 1.3 Research Approach and Objectives

Solving the problem can be done through modeling the problem as an extension of the Multiple Knapsack Assignment Problem (MKAP), a problem that has elements of the traditional Knapsack Problem and the Assignment Problem, with the additional element of multidimensional constraints. These constraints are due to aircraft restrictions on space and weight for not only cargo but also fuel. The first objective is formulating the Multiple Multidimensional Knapsack Assignment Problem (MMKAP). The second objective is to apply the MMKAP to a military airlift problem involving supply and demand bases with realistic cargo demand quantities derived from recent Air Mobility Command reports. The intent of the second objective is to demonstrate the ability of the MMKAP to solve the transportation and assignment problem. The third objective is to ascertain how categorical assumptions

on demand affect aspects of the solution such as aircraft allocation, and the consequences of shorting or exceeding demand in the event of demand mis-estimation.

## **1.4 Organization**

Chapter II reviews the elements of the knapsack problem and its variants, discusses the assignment problem, and explores current methodologies for solving the multiple knapsack assignment problem. Also found in Chapter II are overviews of the Branch-And-Reduce Optimization Navigator and the foundational research to an all-inclusive approach to airlift planning. A methodology for incorporating multiple sources and destinations with stochastic demand into an aircraft assignment model is discussed in Chapter III. Results from analysis on categorical assumptions to mitigate the costs of stochasticity in demand are provided in Chapter IV. Lastly, Chapter V concludes with insights attributed to this project and propositions for further research avenues.

## II. Literature Review

### 2.1 Overview

This chapter discusses the multiple knapsack problem (MKP) and the assignment problem, as well as several hybridizations of exact and heuristic methods to solve each. The traditional knapsack problem (KP) is a combinatorial optimization problem concerned with finding the optimal combination of  $j$  out of  $n$  items with values  $p_j$  and weights  $w_j$ , to fill a “knapsack” to maximize value without busting the knapsack capacity constraint,  $c$  [5].

$$\left\{ \begin{array}{ll} \max & \sum_{j=1}^n p_j x_j \\ \text{subject to} & \sum_{j=1}^n w_j x_j \leq c \\ & x_j \in \{0, 1\}, \quad j = 1, \dots, n \end{array} \right. \quad (1)$$

It is one of Karp’s NP-Complete problems [6]. Approaches that compute optimal solutions for the KP are the use of dynamic programming and the branch and bound method. Both approaches can be time and memory consuming for trivial instances of the KP and are not practical therefore approximation schemes are proposed to solve the problem to optimality or close to optimality while saving time and memory. The KP has many real life applications other than bag packing, such as cutting stock, capital budgeting, and cryptography [7]. Kosuch and Lisser [8] studied a particular instance of a stochastic knapsack where items of unknown weight are assigned to knapsacks in a first stage and can be taken out or added to the knapsack after a second stage, when the actual weights become known. They proved that when searching for good lower bounds, one can replace an exhaustive branch-and-bound framework by a heuristic. Perry and Hartman [9] modeled a case of a stochastic dynamic knapsack,

where items arrive according to a stochastic process and stay in the knapsack for a number of time periods before exiting. Ross and Tsang [10] applied the concept of a stochastic dynamic knapsack to bandwidth allocations of communications switching network to randomly arriving calls of random length.

## 2.2 Multiple Knapsack

The multiple knapsack problem is an extension of the KP problem where there are  $n$  items and  $m$  knapsacks ( $m \leq n$ ). When  $m = 1$ , a MKP reduces to the traditional KP problem [11].

$$\left\{ \begin{array}{l} \max \quad \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \\ \text{subject to} \quad \sum_{j=1}^n w_j x_{ij} \leq c_i, \quad i = 1, \dots, m \\ \quad \quad \quad \sum_{i=1}^m x_{ij} \leq 1, \quad j = 1, \dots, n \\ \quad \quad \quad x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j=1, \dots, n \end{array} \right. \quad (2)$$

The first MKP was published by Eilon and Christofides in 1971 as a loading problem “defined as the allocation of items with known magnitude to boxes with constrained capacity so as to minimize the number of boxes required” [12]. The recommended solutions were to use a zero-one programming model and a heuristic. The heuristic involved a cycle of scanning for items to precisely fill boxes and performed well compared to the zero-one programming method, finding the optimal solution to 48 out of 50 problems in significantly less computing time in comparison. More recently, Lai *et al.* [13] innovated an effective hybrid evolutionary algorithm using a solution-based tabu search for solving the MKP. As of 2017, their algorithm reproduced the best known results for the vast majority of instances tested and established

new best known solutions, or improved lower bounds, for four hard instances. To solve a dynamic, stochastic MKP, Perry and Hartman [9] presented a stochastic dynamic program (SDP) recursion. Their approximation approach utilizes simulation and deterministic dynamic programming to allow for the solution of longer horizon problems and ensure good time zero decisions. Solutions to the MKP can aid in more than answering physical allocation issues; Simon *et al.* [14] applied the MKP to assess the different factors that impact Marine self-sufficiency.

### 2.3 Multidimensional Knapsack

Differing from the multiple knapsack, the multidimensional knapsack problem is an extension of the KP problem where there are multiple resource constraints or a constraint with a multidimensional attribute [5].

$$\left\{ \begin{array}{l} \max \quad \sum_{j=1}^n p_j x_j \\ \text{subject to} \quad \sum_{j=1}^n w_{ij} x_j \leq c_i \quad i = 1, \dots, d \\ \quad \quad \quad x_j \in \{0, 1\}, \quad j = 1, \dots, n \end{array} \right. \quad (3)$$

In 1979, Shih [15] presented a branch and bound method to solve the multidimensional KP. Shih’s branch and bound approach solved thirty problems in 13 minutes compared to 380 minutes using the semi-exhaustive Balas additive algorithm. The Balas additive algorithm uses addition and subtraction to enumerate all  $2^n$  possible solutions of the problem until evidence that no feasible solution exists is obtained or an optimal solution is found [16]. A notable solution method for the multidimensional KP is a hybrid approach based on tabu search by Vasquez and Hao [17]. This method, at the time, improved on the best known results of more than 150 benchmark instances. Recent research by Haddar *et al.* [18] uses an evolutionary

computation technique, the Quantum Particle Swarm Optimization (QPSO), with a local search method to solve the 0-1 multidimensional KP. This produces optimal and near-optimal solutions in a reasonable amount of computational time.

## 2.4 Assignment Problem

A transportation problem has a goal of determining the minimal cost to move a product through a bipartite network to satisfy demands at one half of the network from available supplies at the other half. [19] The assignment problem is a balanced transportation problem where the goal is to minimize the cost of matching supply and demand so that each supply is only matched to one demand and vice versa [20].

$$\left\{ \begin{array}{l} \min \quad \sum_{i=1}^m \sum_{j=1}^n cost_{ij} x_{ij} \\ \text{subject to} \quad \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, m \\ \quad \quad \quad \sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n \\ \quad \quad \quad x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n \end{array} \right. \quad (4)$$

Ram Vaswani [21] originally applied the assignment problem to the allocation of cargo to aircraft. The particular example involved a stipulation that an aircraft must carry no more than one cargo category in the same aircraft. To solve this aircraft assignment problem, Vaswani used the Hungarian Algorithm to assign one of ten types of cargo to one of ten aircraft, with a pre-calculated aircraft-cargo assignment cost provided in matrix form. Ferguson and Dantzig [22] illustrated an application of linear programming to the problem of assigning aircraft to routes to maximize profits and later expanded upon the problem to maximize expected profits when there is uncertain customer demand [23]. Wu and Ross [24] investigated a case of a stochastic assignment problem where balls arrive sequentially (as demands would arrive over a

period of time) and need to be assigned to boxes to minimize arrivals,  $N$ . A ball only fits in certain types of boxes, and the types it can fit in is not known until arrival. The solution to Wu and Ross’s stochastic assignment problem involves a heuristic policy to minimize the expected number of arrivals and a dynamic program to improve upon the heuristic policy.

## 2.5 Multiple Knapsack Assignment Problem

The multiple knapsack assignment problem (MKAP) is an extension of the MKP and of the assignment problem where  $n$  items are broken into  $K$  mutually disjoint subsets of items and the goal is to assign knapsacks to each subset and solve to maximize the total profit of accepted items.

$$\left\{ \begin{array}{l} \max \quad \sum_{i=1}^m \sum_{k=1}^K \sum_{j \in N_k} p_j x_{ij} \\ \text{subject to} \quad \sum_{j \in N_k} w_j x_{ij} \leq c_i y_{ik}, \quad i = 1, \dots, m, \quad k = 1, \dots, K \\ \quad \quad \quad \sum_{i=1}^m x_{ij} \leq 1, \quad j = 1, \dots, n \\ \quad \quad \quad \sum_{k=1}^K y_{ik} \leq 1, \quad i = 1, \dots, m \\ \quad \quad \quad x_{ij}, y_{ik} \in \{0, 1\}, \quad \forall i, j, k \end{array} \right. \quad (5)$$

Kataoka and Yamada [25] first introduced the problem in 2014 and presented a heuristic algorithm to solve this problem approximately as well as three ways to compute the same upper bound. Their solution method for the MKAP uses a greedy heuristic to assign the subsets, performs a local search to improve the results, and then decomposes the problem into  $K$  MKPs. Feasible solutions to the MKPs are truncated to save time and to approximate the lower bound. Dimitrov *et al.* [26] presents special cases of the MKAP regarding emergency relocation of items and the formulation of the MKAP when items to be assigned to knapsacks are identical or of equal value.

## 2.6 Branch-and-Bound Method

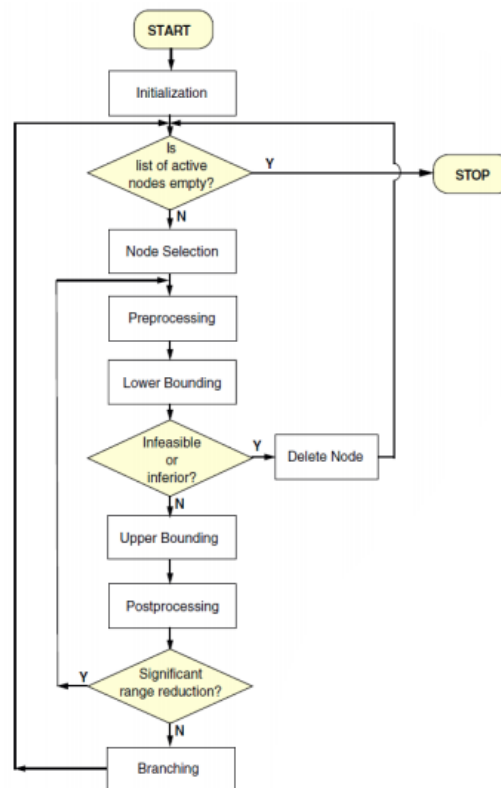
The branch-and-bound method is the repeated partitioning of a solution space into subspaces, solving for the objective value in the subspaces, and decreasing the search space if subspace values are suboptimal. The term *branch* refers to the partitioning of the problem, and *bound* refers to the determination of the solution and therefore the boundary of the subspace. This method utilizes relaxation on constraints to initialize upper and lower bounds for the global optimal solution. If a new branch's solution is outside of these bounds, then it is not considered for further exploration. A successful iterative process of the branch and bound method will converge to a global optimal solution.

William Cook [27] found that the method was proposed in the late 1950s by several individuals. In 1957 Harry Markowitz and Alan Manne described the components of the branch and bound method and a general approach, but did not present an automatic algorithm for solving. Willard Eastman, in his 1958 Ph.D. dissertation (cited in [27]), also used the concept of establishing bounds to eliminate the investigation of branches, however it was not until 1960 that Ailsa Land and Alison Doig [28] presented the numerical algorithm for the branch and bound method. John Little, Katta Murty, Dura Sweeney, and Caroline Karel [29] coined the term “branch and bound” in their 1963 paper on solving the traveling salesman problem.

## 2.7 Branch-And-Reduce Optimization Navigator (BARON)

The branch and bound method is highly utilized today for solving combinatorial and mixed integer problems. Many mathematical program solvers use algorithms of the brand and bound method as the foundation of the program's processing. BARON is one computational system that implements algorithms of the branch-and-bound

type to find the global solution of algebraic nonlinear programs (NLPs) and mixed-integer nonlinear programs (MINLPs). BARON requires bounds on variables and nonlinear expressions in the mathematical program [30], and utilizes feasibility reduction, duality reduction, and a learning heuristic [31] to reduce the range of the solution region until optimality is attained. A high level visual of the BARON algorithm may be seen in Figure 1.



**Figure 1. BARON Algorithm [1]**

To solve the subproblems, BARON calls upon other solvers. By default, BARON utilizes a simplex based optimization software, called CPLEX, for linear problems and the Modular In-core Nonlinear Optimization System (MINOS) for nonlinear problems. CPLEX generally solves linear problems using the dual simplex algorithm. For problems with a nonlinear objective, MINOS solves for local optima using a reduced

gradient algorithm combined with a quasi-Newton algorithm. For problems with non-linear constraints, MINOS solves for local optima by using a projected Lagrangian algorithm. The program iterates through subproblems with linearized versions of the constraints and utilizes the reduced-gradient algorithm to solve [32].

## 2.8 Aircraft Routing and Resources

In [2] Reiman developed regression equations on flight data from aircraft performance manuals to estimate the fuel consumption required of the C-5, C-17, or the C-130 to climb, cruise and descend based on the aircraft gross weight, altitude and distance to travel. These mathematical models, in addition to a nodal reduction heuristic, were utilized to generate fuel efficient route alternatives for the Strategic Airlift Problem (SAP). Follow-on research [4] showed that fuel efficiency and cargo throughput can be improved using this alternative route method and the concept of hopping, compared to current cargo routes. Hopping is when an aircraft makes en-route stops to refuel thus allowing aircraft to carry more in payload weight instead of fuel weight to get to the destination [33]. Extra stops require time, resource availability at enroute airfields of crews and aircraft maintenance, and the premise of hopping relies upon capitalizing on the fuel efficiency of smaller aircraft. While Maywald *et. al.* [4] do not account for resource availability, Baker *et. al.* [34] evaluate the requirements for airfield resources to meet aircraft demand, including aircraft specific maintenance, ramp space and fuel pumping rates.

## III. Methodology

### 3.1 Introduction

This chapter outlines the methodology for the multiple multidimensional knapsack assignment problem approach utilized in this research. Described in depth are the fuel estimation equations and the distance formula used in the model. Next, the mathematical model is outlined. Lastly detailed are how the model inputs are prepared and the solution is processed.

### 3.2 General Mathematical Model

At present, it appears that the Multiple Multidimensional Knapsack Assignment Problem (MMKAP) has not been particularly addressed in literature. The following is a general model for the MMKAP. Given a set of  $I$  supply nodes and  $J$  demand nodes, the objective is to minimize fuel costs to meet the requirements of the respective demand nodes for item types  $K$  by utilizing the set of vehicles,  $L$ , while maintaining the integrity of each vehicle's limitations. The fuel function shown in the objective function, Equation 6, bases fuel consumption on the vehicle, whether the vehicle is assigned, and the total weight transported on the vehicle. The limitations of each vehicle,  $l$ , are represented by a set of multidimensional constraints,  $R$ . Constraining the objective function are the supply inventory and demand requirements, respectively Equations 8 and 7, the dimensional limitations of each vehicle, shown in Equation 9, the assignment constraints that a vehicle can only be assigned once, Equations 10 and 12, and the non-negativity constraint on the items shipped, Equation 13.

**Sets:**

$I$  set of supply nodes, indexed by  $i$

$J$  set of demand nodes, indexed by  $j$

$K$  set of item types, indexed by  $k$

$L$  set of vehicles, indexed by  $l$

$R$  set of dimensions, indexed by  $r$

**Decision Variables:**

$x_{ij}^{kl}$  number of item type  $k$  to transport on vehicle  $l$  from supply node  $i$  to demand node  $j$

$y_{ij}^{kl}$  assignment of vehicle  $l$  to deliver items of type  $k$  from supply node  $i$  to demand node  $j$

**Parameters:**

$D_j^k$  demand of node  $j$  for item type  $k$

$s_i^k$  supply of node  $i$  of item type  $k$

$c_r^l$   $r$ th dimensional constraint for vehicle  $l$

$w_r^k$  size of item type  $k$  in the  $r$ th dimension

$f_{ij}^l$  total fuel consumed in weight units

**Objective Function:**

$$\min \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} f_{ij}^l(l, y_{ij}^{kl}, x_{ij}^{kl}) \quad (6)$$

**Subject to:**

$$\sum_{i \in I} \sum_{l \in L} x_{ij}^{kl} = D_j^k \quad \forall j \in J, k \in K \quad (7)$$

$$\sum_{j \in J} \sum_{l \in L} x_{ij}^{kl} \leq s_i^k \quad \forall i \in I, k \in K \quad (8)$$

$$w_r^k x_{ij}^{kl} \leq c_r^l y_{ij}^{kl} \quad \forall i \in I, j \in J, k \in K, l \in L, r \in R \quad (9)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall l \in L \quad (10)$$

$$\sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall i \in I, j \in J, l \in L \quad (11)$$

$$y_{ij}^{kl} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (12)$$

$$x_{ij}^{kl} \geq 0 \quad \forall i \in I, j \in J, k \in K, l \in L \quad (13)$$

### 3.3 Complexity

The MMKAP is an extension of the Multiple Knapsack Assignment Problem (MKAP), and has at least the same complexity as the MKAP. In addition to the traditional MKAP constraints for the assignment of knapsacks and items, Equations 10, 11, 12, the MMKAP adds multidimensional constraints, Equation 9, demand and supply constraints, Equations 7 and 8, and a non-negativity constraint on the decision variable  $x$  versus the traditional binary constraint, Equation 13. With the addition of these constraints, solution time increases significantly as the problem size increases.

### 3.4 Fuel Estimation

To solve the cargo allocation and transportation problem using the MMKAP, fuel estimation and distance calculations are required. In [2] Reiman developed fuel regression equations based on flight data from aircraft performance manuals for the C-17, C-5, and C-130. These equations model the fuel, in kilo-pounds, required of the three aircraft types to climb, cruise and descend based on the aircraft gross weight, altitude and distance to travel. While the equations allow for a user-defined altitude, the analysis was based on using the optimal cruise altitude for an aircraft's maximum gross takeoff weight. These aircraft assumptions are listed in Table 1.

**Table 1. Aircraft Specific Payload Movement Assumptions [2]**

	C-5	C-17	C-130
Operating Weight $\omega_{op}$	380	282.5	78
Max Gross Takeoff Weight $\omega_{mgt}$	769	585	155
Fuel Capacity $\omega_{fcap}$	347	241.36	62
Aircraft Max Payload, $\omega_{apmax}$	270	170.9	53
Reserve Fuel $\omega_{frc}$	23.45	21.44	4
Alternate Fuel	23.45	21.44	4
Holding Fuel $\omega_{fah}$	17.59	16.08	0
Start Taxi Takeoff Fuel $\omega_{fstto}$	3	4.5	0.67
Approach $\omega_{fapp}$	7	2.67	0.7

Values for each regression model depend on the weights from Table 1 as well as calculated values from one or both of the other regression models. The combination of given weights from Table 1 and the calculated weights result in Equation 14 for the ramp fuel weight. Equation 14 plays a part in Equation 15, the model calculating the fuel required to climb. The regression coefficient values for the climb equation are in Table 2. The regression coefficient values in Table 3 are for Equation 16, the model calculating the fuel required to descend.

$$\omega_{rf} = \omega_{fstto} + \omega_{fc} + \omega_{ff} + \omega_{fd} + \omega_{fapp} + \omega_{frc} + \omega_{fah} \quad (14)$$

$$\omega_{fc} = \beta_0 + \beta_1\alpha + \beta_2\alpha^2 + \beta_3\alpha^3 + \beta_4\omega + \beta_5\omega^2 + \beta_6\omega^3 + 10^{-6}\beta_7\alpha^2\omega^3 + 10^{-6}\beta_8\alpha^2\omega^3 \quad (15)$$

$$\omega_{fd} = \beta_0 + \beta_1\omega_{gd} + \beta_2\omega_{gd}^2 + \beta_3\alpha + \beta_4\alpha\omega_{gd} \quad (16)$$

where:

$\omega_{fc}$  = Fuel to Climb in Klbs

$\omega_{fd}$  = Fuel to Descend in Klbs

$\alpha$  = Altitude in Thousands of Feet

$\omega$  = Aircraft Gross Weight in Klbs at Climb Start

$$= \omega_{rf} + \omega_{op} + \omega_p$$

$\omega_{gd}$  = Aircraft Gross Weight in Klbs at Descent Start

$$= \omega - \omega_{fstto} - \omega_{fc} - \omega_{ff}$$

$\omega_{fstto}$  = Fuel for Start, Taxi, and Takeoff in Klbs

$\omega_{ff}$  = Fuel to Cruise in Klbs

**Table 2. Climb  $\omega_{fc}$  Regression Terms [2]**

	C-5	C-17	C-130
$\beta_0$	-3.0115	-4.7054	-1.067
$\beta_1$	0.3192	0.2869	0.0669
$\beta_2$	-0.0082	-0.007	-0.0022
$\beta_3$	9.50E-05	7.10E-05	3.00E-05
$\beta_4$	0.0164	0.0267	0.0218
$\beta_5$	-3.30E-05	-5.90E-05	-0.0002
$\beta_6$	2.20E-08	4.80E-08	5.20E-07
$\beta_7$	3.70E-05	6.70E-05	0.0003
$\beta_8$	7.10E-08	-2.10E-07	1.30E-05

**Table 3. Descent  $\omega_{fd}$  Regression Terms [2]**

	C-5	C-17	C-130
$\beta_0$	-1.9673	0.2574	-0.0513
$\beta_1$	0.0128	0.0005	-0.0012
$\beta_2$	1.34E-05	-8.50E-07	1.38E-05
$\beta_3$	0.1254	0.0108	0.0367
$\beta_4$	0.0004	3.20E-05	-0.0002

The regression coefficient values for Equation 17, the model calculating the fuel required to cruise, are in Table 4. While the weight of the reserve fuel and the alternate fuel are not planned to be consumed, they are a necessary part in the fuel equations because they add to the overall ramp fuel weight and weight of the plane over the course of the flight.

$$\omega_{ff} = -\frac{B}{3A} - \frac{1}{3A} \sqrt[3]{\frac{1}{2}[2B^3 - 9ABC + 27A^2D + \sqrt{(2B^3 - 9ABC + 27A^2D)^2 - 4(B^2 - 3AC)^3}]} - \frac{1}{3A} \sqrt[3]{\frac{1}{2}[2B^3 - 9ABC + 27A^2D - \sqrt{(2B^3 - 9ABC + 27A^2D)^2 - 4(B^2 - 3AC)^3}]} \quad (17)$$

where (all weights in Klbs):

$$A = \frac{\beta_4}{3}$$

$$B = \left(\frac{\beta_3}{2} + \beta_4(\omega_{op} + \omega_{frc} + \omega_{fah} + \omega_p) + \frac{\beta_5}{2}\alpha\right)$$

$$C = \beta_0 + \beta_1\alpha + \beta_2\alpha^2 + \beta_3(\omega_{op} + \omega_{frc} + \omega_{fah} + \omega_p) + \beta_4(\omega_{op} + \omega_{frc} + \omega_{fah} + \omega_p)^2 + \beta_5\alpha(\omega_{op} + \omega_{frc} + \omega_{fah} + \omega_p)$$

$$D = -\delta$$

$\delta$  = Distance in NMs

$\omega$  = Aircraft Gross Weight

$\omega_{op}$  = Operating Weight

$\omega_{frc}$  = Reserve/Contingency Fuel Weight

$\omega_{fah}$  = Alternate/Holding Fuel Weight

$\omega_p$  = Payload Weight

$\omega_{ff}$  = Fuel to Cruise in Klbs

**Table 4. Cruise  $\omega_{ff}$  Regression Terms [2]**

	C-5	C-17	C-130
$\beta_0$	24.538	31.735	58.829
$\beta_1$	0.5511	0.9897	3.5292
$\beta_2$	0.0002	-0.0043	-0.0098
$\beta_3$	-0.0318	-0.0642	-0.2384
$\beta_4$	0.000019	0.000058	0.001
$\beta_5$	-0.0005	-0.0011	-0.0155

### 3.5 Distance Estimation

The solution to this application of the MMKAP relies heavily on Reiman's [2] fuel estimation equations. One factor in these equations is distance. Utilized in both Reiman's work and in this research is the Vincenty distance formula [35]. Instead of assuming a straight line distance and using the Pythagorean Theorem, or assuming a spherical Earth and using the great-circle distance, the Vincenty Inverse Method assumes the Earth is a flattened spheroid and iteratively calculates distance via ellipsoidal geometry given the latitude and longitude of both the source and the destination points. The reference used for Earth's geospatial properties in Reiman's and this work is the World Geodetic System 1984 (WGS84) [36].

The maximum distance an aircraft can traverse is estimated assuming that cruise and descent fuel have little impact on distance. Equation 17 is used, setting the fuel consumed equal to the maximum amount of fuel an aircraft is allowed to carry given a payload weight, Equation 18, and solving for distance.

$$\mathit{maxFuel} = \mathit{min}(\omega - \omega_{op} - \omega_p, \omega_{fcap}) \quad (18)$$

### 3.6 Applied Mathematical Model

The mathematical program for the problem uses the fuel equations in not only the constraints, but also the objective function, Equation 19. There are two sets of bases, one acting as a supplier and one as the demand. Expected demand is used as the constraint in Equation 20 for the amount required to be sent to the demanding base from all suppliers. The aircraft is a largely constraining factor in the problem. Each aircraft has a maximum weight for payload, takeoff, and fuel, shown in Table 1 and accounted for in Equations 22, 23, 24, respectfully. Aircraft also have a given amount of floor space for cargo [37][38][39], therefore the amount of cargo cannot exceed the given floor space in Equation 25. Cargo is assumed to be on standard pallets, and is constrained to the properties associated with a standard pallet of item type  $K$  in Equation 29; there are no partial pallets. A supplying base has a limit on the number of a specific item type  $K$ , and is only allowed to send what is available in inventory in Equation 21. Equations 26 and 28 are the assignment equations, ensuring an aircraft is only allowed to fly the cargo between one supply-demand pairing.

#### Sets:

$I$  set of supply bases, indexed by  $i$

$J$  set of demand bases, indexed by  $j$

$K$  set of item types, indexed by  $k$

$L$  set of aircraft, indexed by  $l$

#### Decision Variables:

$x_{ij}^{kl}$  number of pallets of item  $k$  to fly on aircraft  $l$  from base  $i$  to base  $j$

$y_{ij}^{kl}$  assignment of aircraft  $l$  to deliver pallets of type  $k$  from base  $i$  to base  $j$

**Parameters:**

- $D_j^k$  expected demand of base  $j$  for item  $k$
- $supply_i^k$  supply inventory of item  $k$  at base  $i$
- $w_k$  weight of item type  $k$
- $\omega_{mgt}^l$  maximum gross takeoff weight for aircraft  $l$
- $\omega_{apmax}^l$  maximum payload weight for aircraft  $l$
- $\omega_{fcap}^l$  maximum fuel weight for aircraft  $l$
- $\omega_{rf}^l$  ramp fuel weight for aircraft  $l$
- $\omega_{frc}^l$  reserve/contingency fuel weight for aircraft  $l$
- $\omega_{fah}^l$  alternate/holding fuel weight for aircraft  $l$
- $f_{ij}^l$  total fuel consumed in Klbs,  $f = \omega_{rf} - \omega_{frc} - \omega_{fah}$
- $space^l$  space capacity of cargo floorspace of aircraft type  $l$
- $fuelPrice$  current market price of fuel per Klb

**Objective Function:**

$$\min \quad \text{fuelPrice} * \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} f_{ij}^l(l, y_{ij}^{kl}, w_k x_{ij}^{kl}) \quad (19)$$

**Subject to:**

$$\sum_{i \in I} \sum_{l \in L} x_{ij}^{kl} = D_j^k \quad \forall j \in J, k \in K \quad (20)$$

$$\sum_{j \in J} \sum_{l \in L} x_{ij}^{kl} \leq \text{supply}_i^k \quad \forall i \in I, k \in K \quad (21)$$

$$\sum_{k \in K} w_k x_{ij}^{kl} \leq \omega_{apmax}^l y_{ij}^{kl} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (22)$$

$$\omega_g^{ijl} \leq \omega_{mgt}^l \quad \forall i \in I, j \in J, l \in L \quad (23)$$

$$\omega_{rf}^{ijl} \leq \omega_{fcap}^l \quad \forall i \in I, j \in J, l \in L \quad (24)$$

$$\sum_{k \in K} x_{ij}^{kl} \leq \text{space}^l y_{ij}^{kl} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (25)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall l \in L \quad (26)$$

$$\sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall i \in I, j \in J, l \in L \quad (27)$$

$$y_{ij}^{kl} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (28)$$

$$x_{ij}^{kl} \text{ integer} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (29)$$

### 3.7 Assumptions

Several assumptions were made in the interest of simplifying the computational complexity of the model. While there are many bases which can act as supply and demand bases, the pavement of supply and demand locations sampled for this problem can handle the take-off and landing of any of the three types of aircraft with their respective maximum payloads. Aircraft maximum payload weight, maximum gross takeoff weight and operating weight are assumed to be fixed as shown in Table 1.

It is assumed airfields are available for routing of aircraft, however, enroute stops are not calculated based on actual base locations, but are equidistant points on the direct route between supply demand pairings. The necessity of one or more enroute stops for a given mission design series (MDS) and payload is calculated in Algorithm 1. The number of stops is determined by dividing the Vincenty distance [35] by the calculated maximum distance the aircraft can fly carrying the payload and rounding this number down to the closest integer. For example, if the distance between the pairings is less than the maximum distance at maximum payload weight, then the ratio would be less than one. This rounds down to a necessity of zero enroute stops. No cargo is delivered or exchanged at enroute stops, however fuel tanks are re-filled to maximum capacity.

---

**Algorithm 1** Number of Stops

---

```
function NUMSTOPS(MDS, distance, payload)
    numberStops = numStops(selectMDS, distance, payload)
    stopDist = maxDistance(payload)
    numStops = roundDown(distance/stopDist)
end function
```

---

Demand bases do not have inventory restrictions and have the equipment necessary to receive and handle any items. Weather is not a limiting factor and is not taken into account for routing or fuel cost. Reserve, contingency, alternate and hold-

ing fuels are safety fuels that are carried but not consumed. The model assumes that aircraft are available and prepositioned to fulfill assignment pairings, and need not fly from another location to get to the supply base. The cost of prepositioning aircraft at supply locations is not a factor. Time is also not a factor. There are adequate crew and aircraft to fulfill the flight requirements of assignments.

### 3.8 General Algebraic Modeling System Implementation

Complications implementing the fuel calculations within the General Algebraic Modeling System (GAMS) resulted in a modification to the mathematical model in the GAMS program. The original objective function is nonlinear in nature and relies on fuel equations that rely not only on the decision variables  $x$  and  $y$ , but also, because the aircraft gross weight fluctuates throughout flight, the fuel equations rely on each other. Due to the complexity of the nonlinear objective function, the objective function that drives the solution in GAMS was altered to solve minimizing an approximation to the fuel cost. Prior to running the GAMS model, the fuel cost of a specific aircraft flying from one supply location to one demand location is calculated for the plane carrying the maximum payload as well as the base fuel cost for flying no payload, Equation 30. The cost per payload kilopound is calculated in Equation 31 by taking the difference between these two values and dividing it by the maximum payload weight for the specific aircraft.

$$baseFuelCost_{ij}^l = f_{ij}^l(l, 1, zeroPayload) \quad (30)$$

$$fuelCost_{ij}^l = \frac{f_{ij}^l(l, maxPayload^l) - baseFuelCost_{ij}^l}{\omega_{apmax}^l} \quad (31)$$

Multiplying the payload by the cost per payload and adding it to the cost of flying the aircraft while empty results in the approximated cost of carrying the payload

that is used in the GAMS objective function, Equation 32. By altering the objective function from Equation 19 to Equation 32, the GAMS model no longer requires constraint Equations 23 or 24. Finally, to focus on the aspect of demand in the analysis, supply is assumed to be unlimited, therefore the supply constraint, Equation 21, is also removed.

$$\min \quad fuelPrice * \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (w_k x_{ij}^{kl} * fuelCost_{ij}^l + baseFuelCost_{ij}^l) \quad (32)$$

**Subject to:**

$$\sum_{i \in I} \sum_{l \in L} x_{ij}^{kl} = D_j^k \quad \forall j \in J, k \in K \quad (20)$$

$$\sum_{k \in K} w_k x_{ij}^{kl} \leq \omega_{apmax}^l y_{ij}^{kl} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (22)$$

$$\sum_{k \in K} x_{ij}^{kl} \leq space^l y_{ij}^{kl} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (25)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall l \in L \quad (26)$$

$$\sum_{k \in K} y_{ij}^{kl} \leq 1 \quad \forall i \in I, j \in J, l \in L \quad (11)$$

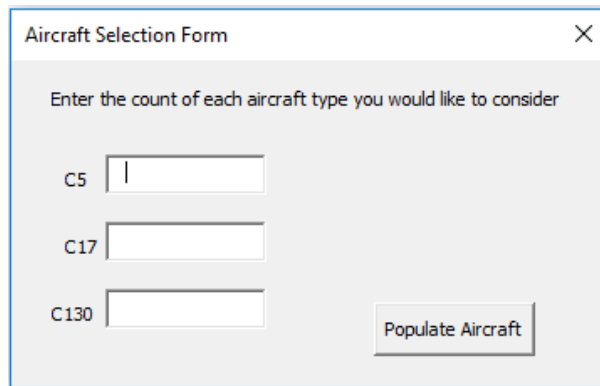
$$y_{ij}^{kl} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (28)$$

$$x_{ij}^{kl} \text{ integer} \quad \forall i \in I, j \in J, k \in K, l \in L \quad (29)$$

### 3.9 Solution Pre-Processing

Many of the subroutines and functions, written in Microsoft Excel Visual Basic for Applications (VBA), are used for data collection and pre-processing. Initially, the user is prompted to select the supply and demand airfields from a list of 5342 airfields from the Digital Aeronautical Flight Information File (DAFIF) database.

The user may manually add the specific International Civil Aviation Organization (ICAO) airport code to the supply base list, or they may use a button to populate the list with the known channel continental United States (CONUS) supply bases, or, for completely proof of concept purposes, the user may use a button to randomly populate the list with a specific number of supply bases. With the selection of airfields the user is requested to input the inventory of the supply bases and the requirements for the demand bases of two cargo types, high priority, “Priority 1” or “Super/999/1,” cargo and regular priority, “Priority 2/3” cargo. They may do this manually, or may use a button on the respective pages to randomly populate the supply and demand. Up to this point, a supply sheet and a demand sheet have been created. Lastly, the user is asked to add the number of available C-5Bs, C17s, and C-130J-30s via the aircraft selection form, shown in Figure 2.



Aircraft Selection Form

Enter the count of each aircraft type you would like to consider

C5

C17

C130

Populate Aircraft

**Figure 2. Aircraft Selection Form**

Upon clicking the “Populate Aircraft” button, many subroutines and functions are called upon to create the aircraft page. This lists available aircraft, their associated parameters such as maximum takeoff weight and fuel capacity, and the calculated aircraft costs per payload and baseline fuel costs to fly between the source and destination pairings. The main subroutine called to perform this page creation is called GetDistance. As seen in Algorithm 2, the GetDistance subroutine initially finds the

latitude and longitude for the supply base, then iterates through the demand bases. For each demand base, the latitude and longitude is found, then the program iterates through all available aircraft. The `vincentyDistance` function is called to calculate the Vincenty distance between the supply and demand base and fuel function is called to calculate the fuel cost given the distance, MDS, and payload. For the base fuel calculation the payload is zero. For the max fuel calculation the payload is the maximum payload for the given aircraft type.

---

**Algorithm 2** Sub Routine GetDistance

---

```

for each supply base  $i \in I$  do
  for all airfields do
    if airfield = supplybase then
       $lat1 = airfield.latitude$ 
       $long1 = airfield.longitude$ 
      exit for all airfields
    end if
  end for
  for each demand base  $j \in J$  do
    for all airfields do
      if airfield = supplybase then
         $lat2 = airfield.latitude$ 
         $long2 = airfield.longitude$ 
         $distance = vincentyDistance(lat1, long1, lat2, long2)$ 
        for all aircraft  $l \in L$  do
           $MDS = aircraft.MDS$ 
           $maxPayload = aircraft.maxpayload$ 
           $baseFuel = fuel(MDS, distance, payload=0)$ 
           $payloadFuel = fuel(MDS, distance, payload=maxpayload)$ 
           $maxFuel = (payloadFuel - baseFuel)/maxPayload$ 
          write supplyBase, demandBase, baseFuel, maxFuel
        end for
      exit for all airfields
    end if
  end for
end for
end for

```

---

As seen in Algorithm 3, the fuel function calls upon the `numStops` function to

provide the number of stops the aircraft requires given payload and distance and calls upon the `fuelCalculation` function to calculate the amount of fuel consumed for a given aircraft type, distance, number of stops, and payload amount. The `fuelCalculation` function calculates fuel consumption using the fuel equations detailed in Section 3.4. All functions and subroutines may be found in Appendix A.

---

**Algorithm 3** Fuel Calculations

---

```

function FUEL(MDS, distance, payload)
    numberStops = numStops(MDS, distance, payload)
    fuel = fuelCalculations(MDS, distance, payload, numberStops)
end function

```

---

### 3.10 Solution Processing

GAMS Data eXchange (GDX) facilities pull data and parameters from the Excel workbook into GAMS. After the pre-processing detailed in Section 3.9, the Excel workbook provides GAMS with the sets of supply bases and demand bases with their respective associated inventory and demand, and the set of available aircraft, with their associated parameters of operating weight, fuel capacity, maximum gross takeoff weight, average altitude, maximum payload capacity, and the aircraft’s baseline cost and cost per payload pound for all pairings of supply and demand bases. The modified model from Section 3.8 is solved in GAMS for a nonlinear mixed integer program using BARON [40]. The GAMS code may be found in Appendix B. Once solved, GDX writes the aircraft assignments, payload weights, cargo allocations, and demand base requirements shortage and overage to sheets in the same Excel workbook from which the data is called. The `finalCost` subroutine, Algorithm 4, then calls upon the `fuel` function and uses these outputs to calculate the final total cost.

---

**Algorithm 4** Sub Routine finalCost

---

```
for each Supply-Demand base combination do
  for each allocated aircraft do
    MDS = aircraft.MDS
    distance = myDistance(supplyBase, demandBase)
    payload = aircraft.finalpayload
    finalFuel = fuel(MDS, distance, payload) + fuel(MDS, distance, 0)
    write supplyBase, demandBase, finalFuel
  end for
end for
```

---

### 3.11 Demand

Expected demand derives from real world data reported by the 618th Air Operations Center (AOC) [41][42]. For the purposes of this research, monthly shipment data over the 2017 and 2018 fiscal years (FY) is extracted. Among the details in this data are the origination and destination bases and the tonnage of each of Super/999/1 and Priority 2/3 goods shipped between the given bases over the span of a calendar month. To derive the expected monthly demand for pallets of Super/999/1 and Priority 2/3, the tonnage shipped is assumed to be the demand of a particular base. The number of pallets is calculated by dividing the tonnage by the reported FY 2017 average pallet weight of 1.3 tons per pallet. A daily pallet demand is calculated by dividing the monthly pallet demand by the number of days in the given month. A weekly demand is calculated by multiplying the daily demand by seven. The monthly demand over two fiscal years for the number of selected channels are combined to inform the expected demand of the different priority items. These demand lists are used in two different ways. Firstly, quantiles are computed for the priority types. The model is run with the expected demand set at the different combinations of the demand quantiles. Secondly, triangular distributions are created. Random draws from these distributions are used to actualize the stochastic demand for post solution processing and assessment.

## IV. Analysis

### 4.1 Introduction

A notional example with two supply bases and two demand bases is utilized for analysis. The model is initially solved given a specific expected demand for each priority type. With two priority types, Priority 1 or “Super/999/1” and Priority 2/3, and quantiles of real world expected demand, there are 25 expected demand combinations for which the model is run. Post-solution analysis ascertains how categorical assumptions on demand affect aircraft allocation and assess the effects of stochastic demand on monetary penalties associated with shorting or exceeding demand in the event of mis-estimation.

### 4.2 Pre-Analysis Processing

Considered for analysis are the transports from the continental United States (CONUS) to anywhere else in the world (OCONUS). Channels with highest total shipping volume deriving from each of the East and West coasts of CONUS are McGuire AFB (KWRI) to Ramstein AB (ETAR) and Travis AFB (KSUU) to Osan AB (RKSO). Due to the high volume of shipments over these channels, analyses of the expected demand is on the weekly demand. A combined list of demand over the 2017 and 2018 fiscal years (FY) for KWRI to ETAR and KSUU to RKSO results in 44 data points per priority type. There are four less data points per priority type than expected because shipments between McGuire AFB and Ramstein AB were not reported in FY 2018 April-July. The quantiles for the respective demand of each priorities can be seen in Table 5.

**Table 5. Quantiles for Expected Demand**

Percentile	Super/999/1	Priority 2/3
0th	22	13
25th	50	21
50th	80	30
75th	106	35
100th	155	49

Aircraft availability is determined by the number of each aircraft type necessary to carry the maximum cargo quantiles to each selected OCONUS location. In this instance, there is a maximum total demand across OCONUS bases of 310 pallets of Super/999/1 cargo and 98 pallets of Priority 2/3 cargo or a total of 408 pallets and 1060.8 Klbs. Between space capacity and weight capacity, the limiting factor for number of aircraft is space capacity. For the notional example, available aircraft are set as the number of aircraft to ship the maximum average demand level for both priority types for one week. The aircraft requirements of each MDS for the maximum demand level are 14 C-5s, 24 C-17s, or 52 C-130s. The GAMS model is ran 25 times, each instance solving for a different expected demand quantile combination. After each run, aircraft and cargo allocations are saved in an Excel file and a final cost for the demand combination is computed. The final result is a set of 25 point estimates for cost given differing expected demand amounts for two cargo types. Figure 3 demonstrates the difference between the cost derived from using a fuel approximation in GAMS and the final cost computed in post solution processing.

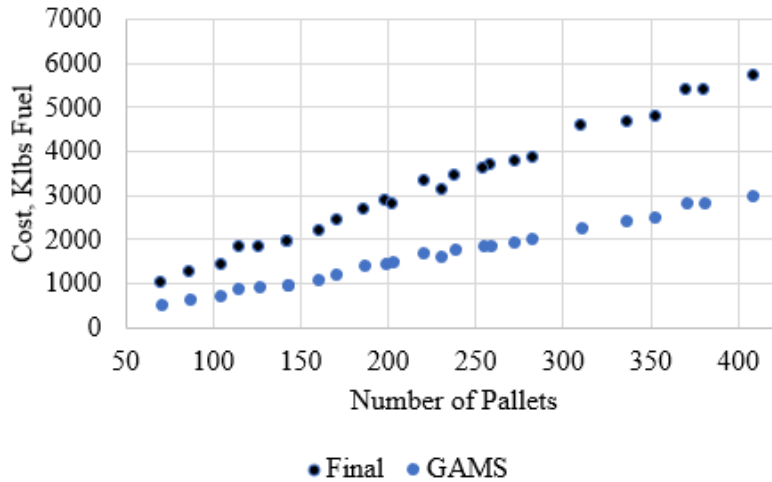


Figure 3. Graph Comparison of GAMS and Post-Processed Final Cost vs. Number of Pallets

### 4.3 Actualizing Demand

After model processing an initial cost frontier is created. Figure 4 shows that as pallet number increases, the cost converges to near 14 Klbs of fuel per pallet.

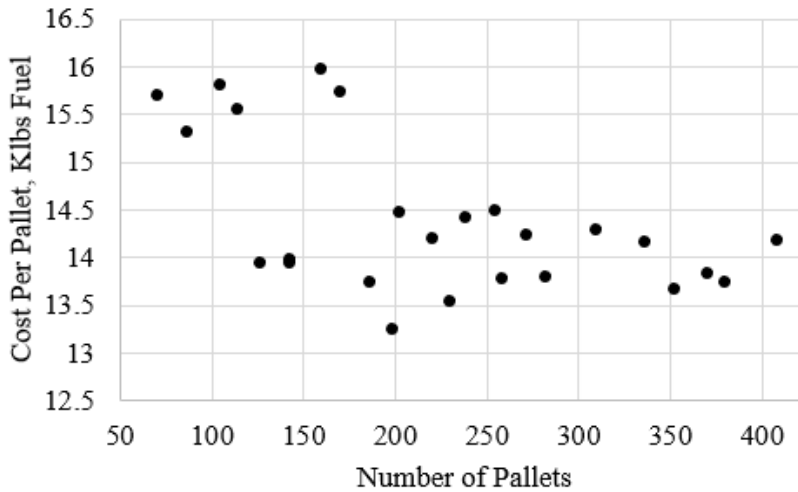


Figure 4. Graph Cost per Pallet vs. Number Pallets: No Priority Cost

Of note, the lower pallet amounts result in the highest cost per pallet. A simple explanation for this is the under-utilization of aircraft with lower pallet amounts.

#### 4.4 Aircraft Allocation

No C-17s are utilized to transport cargo from CONUS to the chosen OCONUS bases of Ramstein AB and Osan AB. This is likely due to the similar fuel efficiency between the C-5 and C-17, and due to the fact that the C-5 has a higher cargo throughput compared to the C-17 when not accounting for mission capable rates [2]. Due to the lighter pallet weight of 2.6 Klbs, the C-130 is more fuel efficient compared to the C-17 and C-5. To determine the near-optimal number of each aircraft to fly the quantile combinations, the model was run for each demand combination utilizing a single type of MDS. Due to the fuel approximation technique used to run the GAMS model and the optimization criterion set for the BARON solver, the solutions are within a 10 percent tolerance of the optimal solution, but not necessarily optimal, and result in the mixed solution not always besting or matching the single MDS solutions. The fuel approximation drives the solution to add C-130s instead of C-5s or C-17s. The optimality criteria of the model also results in a solution that is within 10 percent of optimality, so there is a margin of error in both the mixed and single MDS solutions that makes it difficult to strictly compare them. Ideally, the mixed solution should always be either better or equivalent to the single MDS solutions, as the single MDS solution is a special case of a mixed solution. For the notional example, the mixed solution is less costly 80 percent of the time compared to C-130s and 84 percent with C-5s. The mixed model is cheaper than the C-17 100 percent of the time. To exemplify the aforementioned set-backs due to model optimality criterion and the fuel approximation, Figure 5 shows that with a demand of 272 pallets, the C-5 only solution of using eight aircraft bests the mixed solution of using six C-5s and 10 C-130s. Re-solving the problem with a mixed profile and an optimality criterion of one percent results in a solution that removes a C-5 and replaces it with three C-130s.

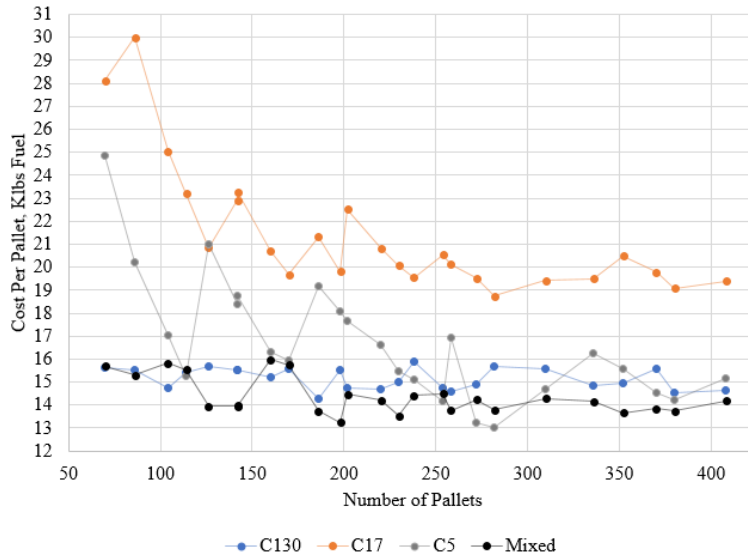


Figure 5. Cost Per Pallet vs. Number of Pallets: Comparison of MDS Only and Mixed Profiles

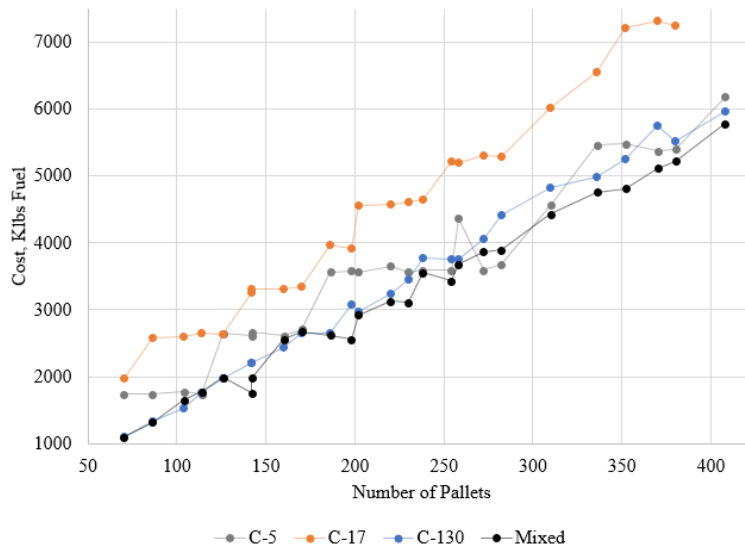
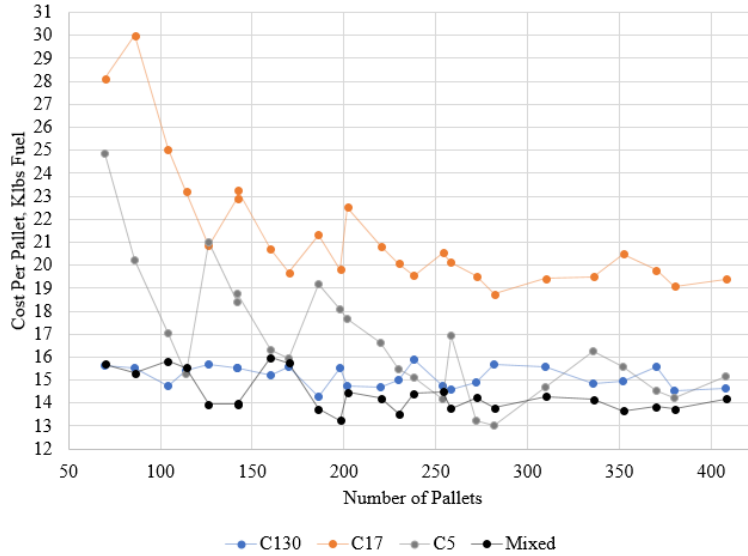
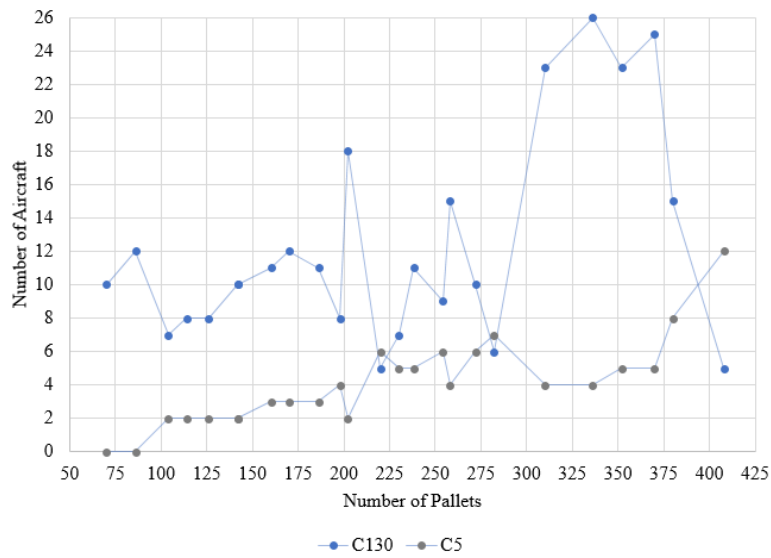


Figure 6. Graph Final Cost vs. Number of Pallets, Separate MDS Solution

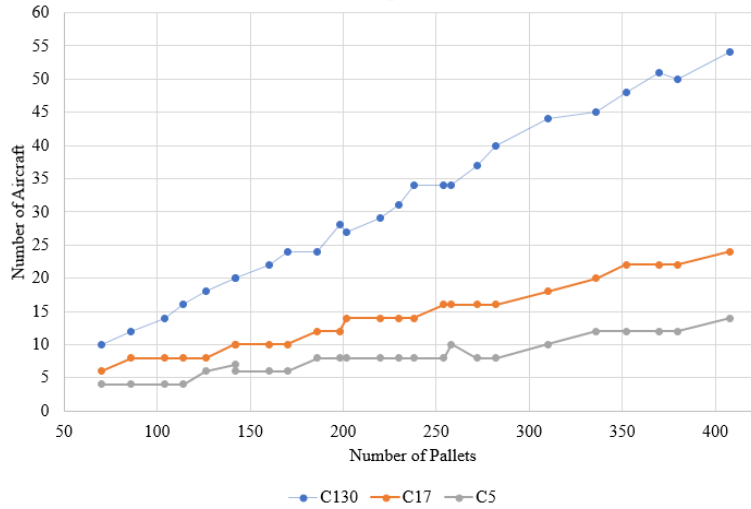


**Figure 7. Graph Cost Per Pallet vs. Number of Pallets, Separate MDS Solution**

The number of aircraft allocated to fly for a given number of pallets are depicted in Figure 8 for the mixed solution and Figure 9 for the single MDS solutions. A comparison of the mixed solution and the single MDS solutions reveals that a mixed solution uses less aircraft than allocating a single MDS, which may be a more realistic depiction of available resources.

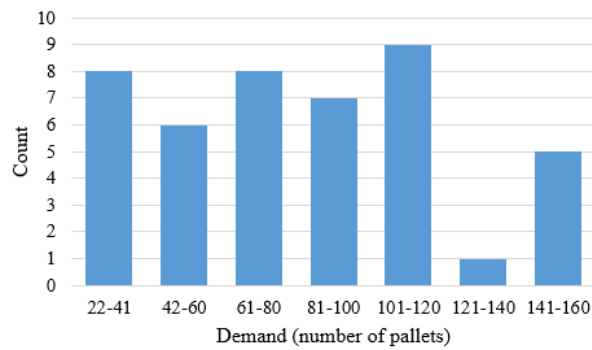


**Figure 8. Number of Aircraft vs. Number of Pallets: Mixed Profile**

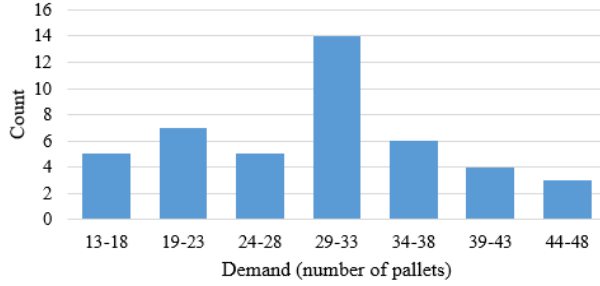


**Figure 9. Number of Aircraft vs. Number of Pallets: Separate Profiles**

To assess stochasticity in demand, a cost frontier for the 25 point estimates is created that accounts for an “actual” demand that is realized post solution processing. Graphing the demand data suggests that the Super/999/1 cargo may be modeled by a uniform distribution, while the Priority 2/3 cargo follows a triangular distribution.



**Figure 10. Histogram of Super/999/1 Cargo**



**Figure 11. Histogram of Priority 2/3 Cargo**

Demand distributions are created using data parameters for each of the cargo types, found in Table 6. One hundred demand samples are populated through random draws from the corresponding distributions.

**Table 6. Data Parameters per Priority Type**

	Super/999/1	Priority 2/3
Min	22	13
Mode	N/A	32
Max	155	49

The amount of excess and shortage of cargo of each type is calculated and multiplied by a priority cost depending on type of offense (excess or shortage) and cargo type. Several priority cost schema are used since “real world” cost consequences are not documented for the generic cargo descriptions of Super/999/1 or Priority 2/3.

#### 4.5 Fixed Cost Models

The “Fixed” penalty cost model sets a fixed cost, in units of kilo-pounds jet fuel, for shorting or exceeding the actual demand for each cargo priority type. The first penalty cost allotment penalizes exceeding the actualized demand and does not set a penalty cost for shorting demand. In this model, the least costly demand combination given an interest in minimizing overage depends on the amount of the fixed penalty cost. Figure 12 compares the cost per pallet of three least costly demand quantile

combinations over the range of fixed costs. Figure 13 depicts the expected demand settings, depending on the penalty amount, that would minimize the expected cost per pallet in the event of a stochastic demand. In this case, the best expected demand settings for an overage penalty cost from zero to 16 Klbs fuel would be the 0th percentile for Super/999/1 cargo and the 100th percentile for Priority 2/3 cargo. The magnitude of demand is larger than that of Priority 2/3 cargo, and might influence the Priority 2/3 cargo demand setting to be at the 100th percentile. This would be a tradeoff on penalty costs for the smaller demand and per pallet cost savings on bulk shipping. Once the fixed overage cost per pallet is 18 Klbs of fuel, the expected demand settings for both cargo types change to the 25th percentile.

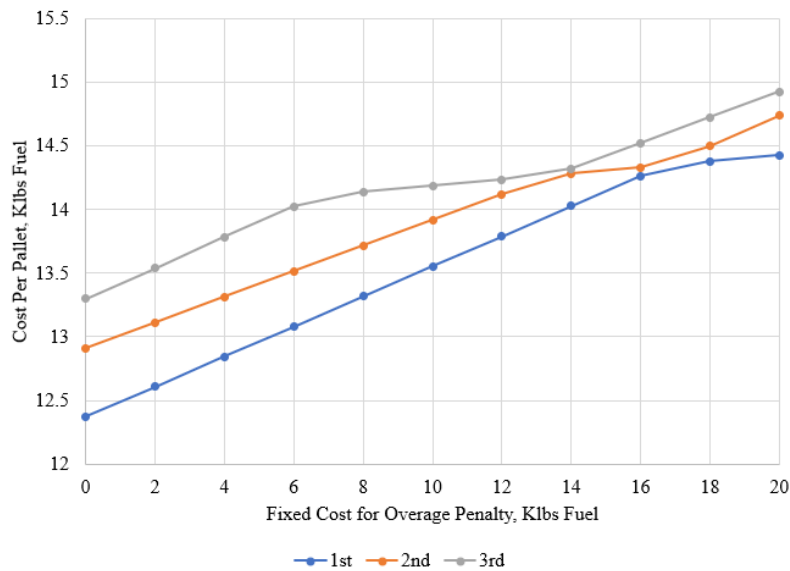
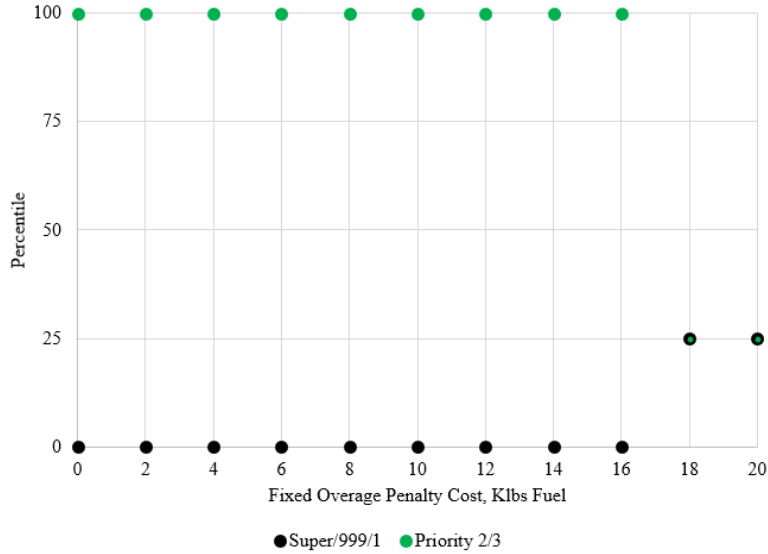


Figure 12. Cost Per Pallet vs. Fixed Penalty Cost for Overage



**Figure 13. Percentile vs. Fixed Penalty Cost for Overage**

The second fixed penalty cost model is arranged so that the penalty costs are associated with shorting the actualized demand. There are no penalty costs for exceeding demand. Figure 14 shows that the cost per pallet of least costly demand quantile combinations is more apparent when the fixed cost for shortage is greater than 10 Klbs of fuel. Figure 15 depicts that the same percentile settings that produce the smallest cost per pallet after a fixed cost of eight kilo-pounds of fuel. These are setting the Super/999/1 cargo expected demand to the 100th percentile and Priority 2/3 expected demand to the 75th percentile.

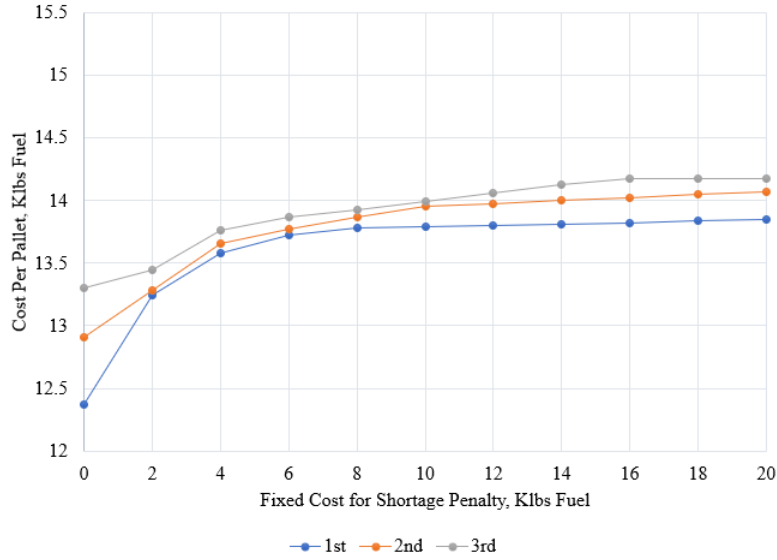


Figure 14. Cost Per Pallet vs. Fixed Penalty Cost for Shortage

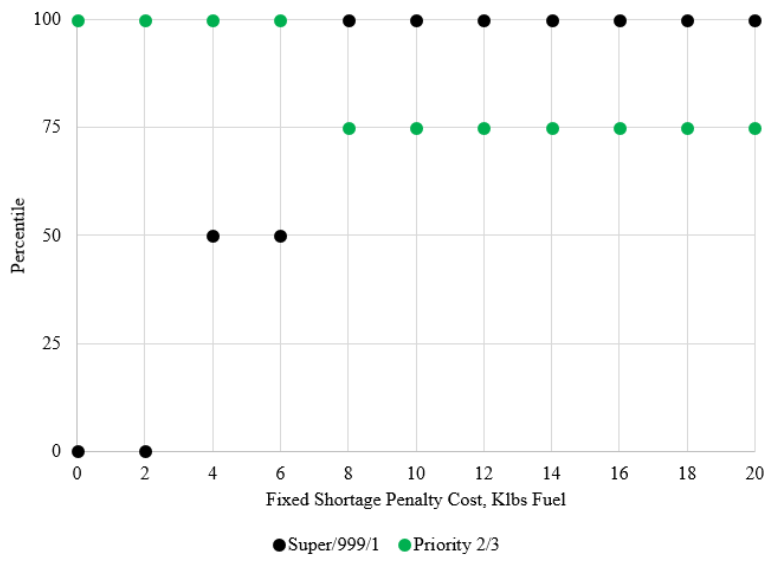


Figure 15. Percentile vs. Fixed Penalty Cost for Shortage

The final fixed cost model equally penalizes exceeding and shorting actualized demand. Figure 16 shows that the top three contending percentiles for the given penalty costs are not distinct from each other and Figure 17 does not depict a clear percentile demand setting that would perform well over a large range of penalty costs.

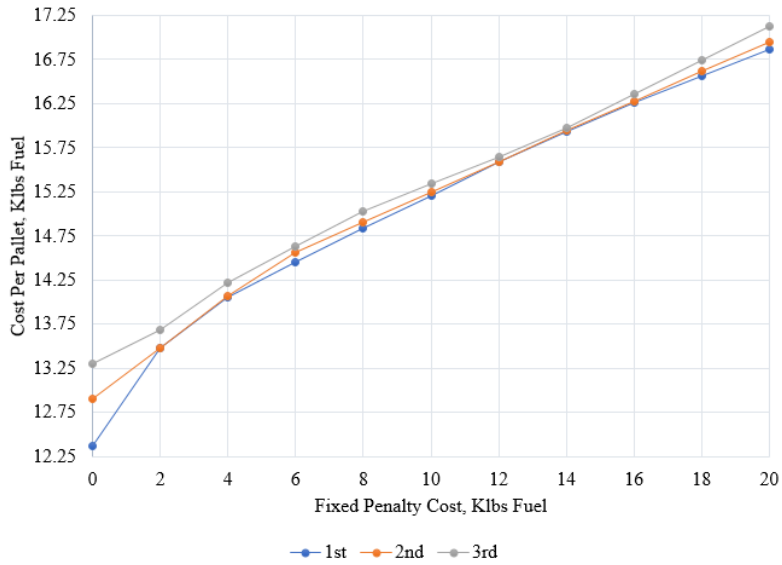


Figure 16. Cost Per Pallet vs. Fixed Penalty Cost

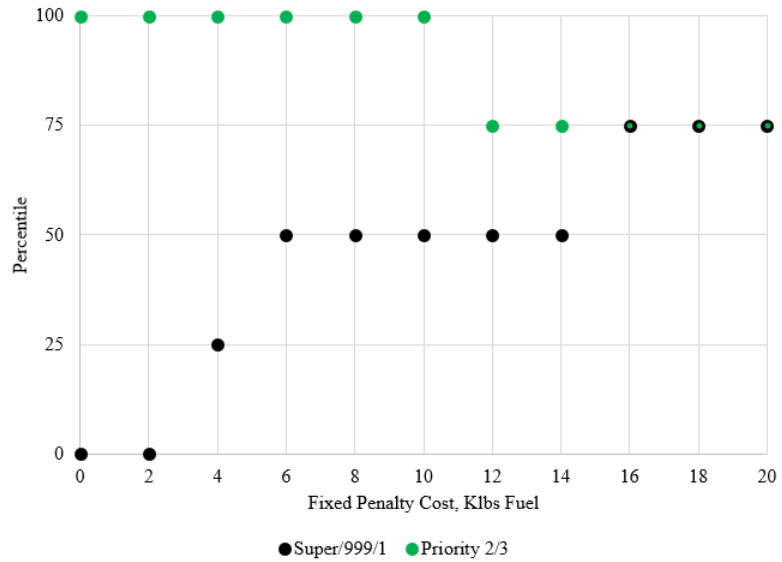
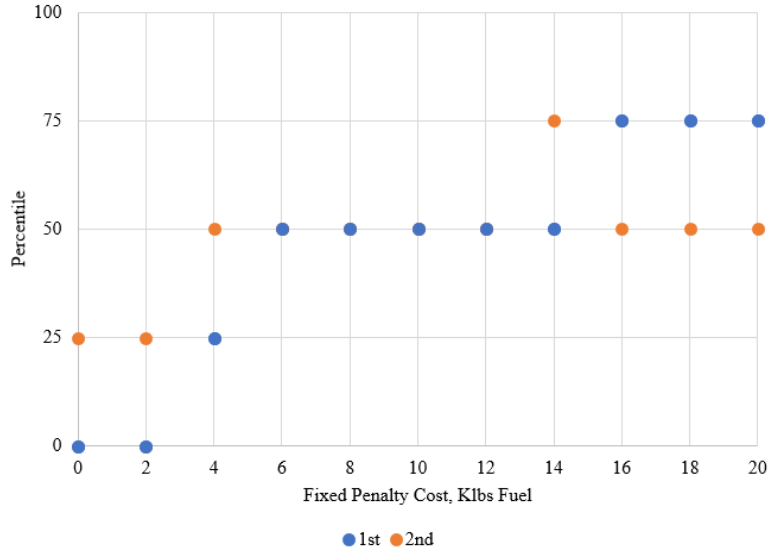


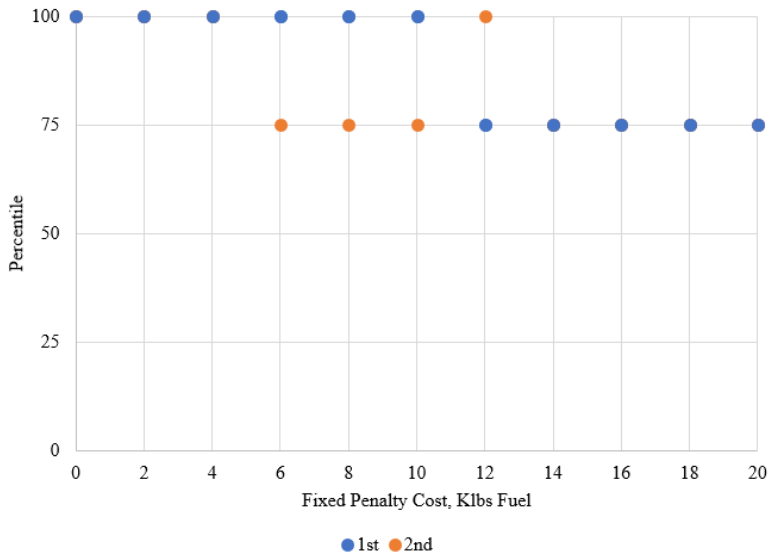
Figure 17. Percentile vs. Fixed Penalty Cost

Since the top two contending settings in Figure 16 are intermingled for a great portion of the penalty costs, the top two contenders were analyzed further. Figure 18 shows a comparison of the top two demand settings for Super/999/1 cargo. This shows that a possible expected demand setting minimizing most penalty costs would be setting Super/999/1 cargo expected demand to the 50th percentile.



**Figure 18. Percentile vs. Fixed Penalty Cost: Super/999/1 Cargo**

Figure 19 shows a comparison of the top two demand settings for Priority 2/3 cargo. This shows that a possible expected demand setting that would minimize most penalty costs would be setting expected demand to the 100th percentile for lower penalty costs up to 12 Klbs of fuel, with the option of setting the demand to the 75th percentile for penalty costs from 6 Klbs to 20 Klbs of fuel.



**Figure 19. Percentile vs. Fixed Penalty Cost: Priority 2/3 Cargo**

## 4.6 Variable Cost Model

The original pallet cost is the fuel amount required to fly aircraft with a specified cargo amount from a source to a destination and return to the source empty. Taking into consideration that over-shipping an item may require returning the overage amount, the penalty cost can be viewed as a percentage of the cost to transport the pallet. The penalty cost could be a range of percentages to account for the return of the pallet. The portion of flight costs associated with transporting cargo ranges from approximately 50 to 57 percent of the original pallet transportation cost.

Figure 20 depicts the three least costly demand settings for the variable cost model where the penalty cost amount is set to 100 percent for overage. The only variable cost model analyzed is on overage, based on the assumption that the cost would be either an inconvenience fee for the destination to hold the excess cargo or the resulting cost to return the excess to the source. Shortage would be a separate order cost and not part of the penalty associated with this order. This model shows that the per pallet cost can be minimized by setting Penalty 2/3 cargo to the 75th percentile when the penalty costs are between 10 and 100 percent. Super/999/1 cargo expected demand settings are less standardized throughout the range of penalty costs, and trend toward lower percentiles as the penalty cost percentage increases.

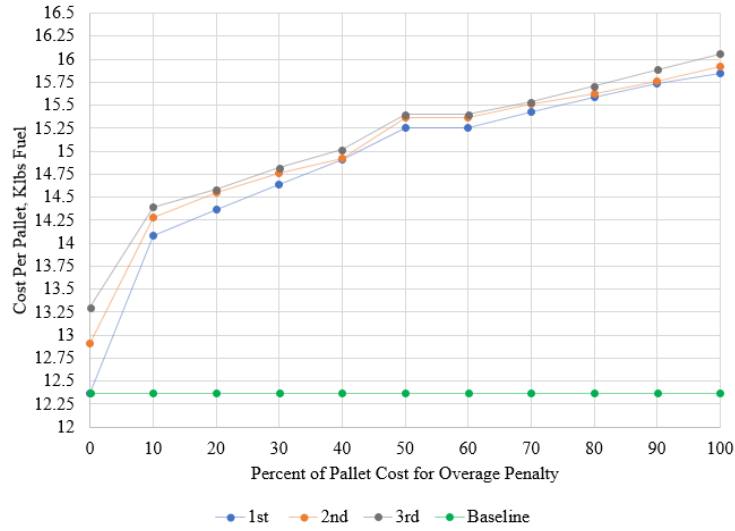


Figure 20. Cost Per Pallet vs. Percent of Pallet Cost for Overage Penalty

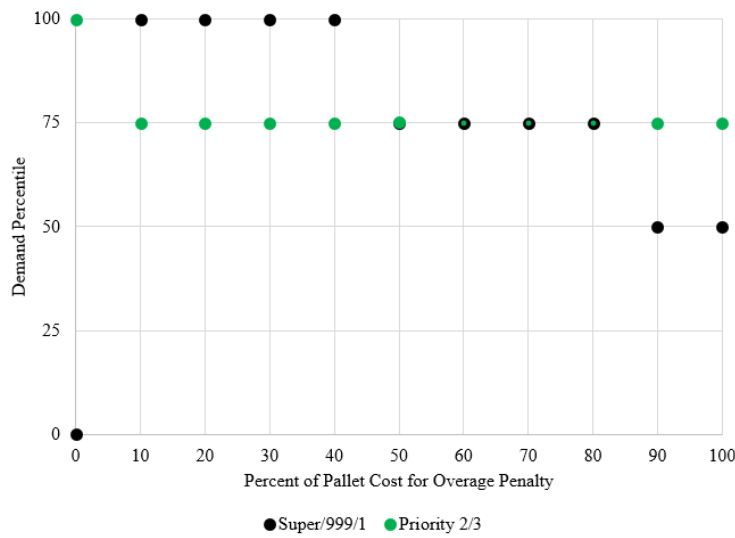


Figure 21. Percentile vs. Percent of Pallet Cost for Overage Penalty

## V. Conclusions and Future Research

### 5.1 Conclusion

This research developed and analyzed a new variant of the MMKAP and showed its applicability on a notional network populated with historical demand signals. The formulation of the MMKAP differed from the MKAP with the addition of multidimensional constraints as well as a constraint to meet expected demand, and a constraint to stay within supply limits. Another difference from the MKAP to the MMKAP was the alteration of the decision variable  $x$  to be non-negative instead of binary. The model was solved deterministically for 25 different demand combinations based on the quantiles of reported cargo data. The computational time to solve the problem was short for most instances of the problem, but this is due to the small size of the example. Brief experimentation with increases in the number of supply bases, demand bases, and aircraft demonstrated a rapid growth in computational time, indicating the intractability of the MMKAP without the use of a heuristic.

To find out how categorical assumptions about demand affect aircraft allocation, the model was run again for the demand quantiles, setting the available aircraft to single MDS types. Allocated aircraft in the original mixed model were only C-5s and C-130s. This may be due to the lighter pallet weight and similar fuel efficiency of the C-5 and the C-17 but dissimilar cargo throughput, or may be the result of the fuel approximation used to run the GAMS model. It was noted that with tighter optimality criterion, the GAMS model favored utilizing C-130s when the lower cost option would have been utilizing a C-5.

The effects of stochasticity of demand on costs were assessed by drawing 100 random actualized demands for each priority type given assumed distributions based off of reported cargo data. Analysis on penalizing the differences of the categorical

assumptions from the actualized demand showed that there are expected demand allocations that minimize penalty cost over a range of demands depending on cost profile and penalty prioritization. From this research, a policy to source a certain level of expected demand that minimizes future expected costs can be identified.

## 5.2 Future Research

Considerations should be taken to improve the model by incorporating the fuel equations as well as programming the enroute bases in the solution processing to give a more accurate fuel estimation. These implementations, due to the nonlinear nature of an objective function based on the fuel equations, would most likely require a better understanding of the functions and limitations of GAMS and VBA. Research with larger problems may be able to leverage the MMKAP if a heuristic, like a tabu search algorithm or a genetic algorithm, is applied. Finally, real world transportation problems hinge on time. An applicable problem would be to investigate the time effects on cost with a stochastic demand such that there is a tradeoff between fulfilling a demand now, with the possible loss of productivity of an aircraft, versus saving capacity for the arrival of future demands. Saving capacity for future arrivals would maximize the productivity of the aircraft but would not fulfill all demands expediently. The priority of the demand, positioning of assets, capacity of aircraft, and transportation cost would all affect this decision.

# Appendix A. Visual Basic for Applications Code

## 1.1 Fuel

### Option Explicit

Public plane, C5, C17, C130, fromBase, toBase As Long

Public selectMDS As String

Function fuelCalculations(selectMDS, distance, Payload, numStops)

'Find the cost of fuel for the max payload

Dim specRange, wgross, gwForMaxDist, fuel, wff, wfc, wfd, A, B, C, D, \_  
fuelReq, myAnswer As Double

Dim i, j, num As Long

Dim plane, wop, wgmt, wmaxfuel, wpmax, wfrah, wfstto, wfapp, wp, alt As Double

Dim distRange, climb, cruise, descent, given As Range

Set distRange = Range("rangeBeta")

Set cruise = Range("cruiseBeta")

Set climb = Range("climbBeta")

Set descent = Range("descentBeta")

Set given = Range("GivensTable")

plane = getPlane(selectMDS)

'for ease of reading the equations.

wop = given(plane, 2)

wgmt = given(plane, 3)

wpmax = given(plane, 5)

wfrah = given(plane, 6)

wfstto = given(plane, 7)

wfapp = given(plane, 8)

alt = given(plane, 9)

wmaxfuel = wgmt - wop - wfrah - Payload

If wmaxfuel + wfrah > given(plane, 4) Then wmaxfuel = given(plane, 4) - wfrah

wgross = wop + Payload + wmaxfuel + wfrah 'gross takeoff weight

num = numStops + 1

'A-D are used in the fuelConsumed Function, found under Equation 23 from

'Reiman 2014 page 52

```

A = cruise(plane, 6) / 3
B = cruise(plane, 5) / 2 + cruise(plane, 6) * (wop + wfrah + Payload) + -
(cruise(plane, 7) / 2) * alt
C = cruise(plane, 2)+cruise(plane, 3)*alt+cruise(plane, 4)*(alt)^2 + -
cruise(plane, 5) * (wop + wfrah + Payload) + -
cruise(plane, 6) * (wop + wfrah + Payload)^ 2 + -
cruise(plane, 7) * alt * (wop + wfrah + Payload)
D = -distance / num

wff = fuelConsumed(A, B, C, D)

'Equation 19 and 20 from Reiman 2014 page 50
wfc = climb(plane, 2) + climb(plane, 3) * alt + climb(plane, 4) * (alt)^ 2 + -
climb(plane, 5) * (alt) ^ 3 + climb(plane, 6) * wgross + -
climb(plane, 7) * (wgross) ^ 2 + climb(plane, 8) * (wgross) ^ 3 + -
climb(plane, 9) * 10 ^ (-6) * (alt) ^ 2 * (wgross)^ 3 + -
climb(plane, 10) * 10 ^ (-6) * (alt) ^ 2 * (wgross)^ 3
wfd = descent(plane, 2)+descent(plane, 3)*(wgross-wfc-wff-wfstto) + -
descent(plane, 4)*(wgross-wfc-wff-wfstto)^ 2 + descent(plane, 5)*alt + -
descent(plane, 6) * alt * (wgross - wfc - wff - wfstto)

fuel = num * (wff + wfc + wfd)

fuelCalculations = fuel

```

**End Function**

'this function returns the Klbs fuel consumed during the cruise portion of flight

**Function** fuelConsumed(A, B, C, D) As Double

**Dim** commonTerm1, commonTerm2, cubeRoot1, cubeRoot2 As Double

```

commonTerm1 = 2# * (B ^ 3) - 9# * A * B * C + 27# * (A^ 2) * D
commonTerm2 = 4 * ((B ^ 2) - 3 * A * C)^ 3
cubeRoot1 = 0.5 * (commonTerm1 + Sqr((commonTerm1^ 2) - commonTerm2))
cubeRoot2 = 0.5 * (commonTerm1 - Sqr((commonTerm1^ 2) - commonTerm2))

```

**If** (cubeRoot1 < 0) **Then**

cubeRoot1 = -(-cubeRoot1)^ (1 / 3)

**Else**

cubeRoot1 = (cubeRoot1^ (1 / 3))

**End If**

```

If (cubeRoot2 < 0) Then
    cubeRoot2 = -(-cubeRoot2)^(1 / 3)
Else
    cubeRoot2 = (cubeRoot2^(1 / 3))
End If

'//g=-B/3A
'// -1/3A(1/2 [2B^3-9ABC+27A^2 D+v((2B^3-9ABC+27A^2 D)^2-4(B^2-3AC)^3)])
'// -1/3A(1/2 [2B^3-9ABC+27A^2 D-v((2B^3-9ABC+27A^2 D)^2-4(B^2-3AC)^3)])

fuelConsumed = (-B/(3*A)) - (1/(3*A)) * cubeRoot1 - (1/(3*A)) * cubeRoot2

```

**End Function**

```
'This function facilitates the fuel costs (accounts for multiple enroute stops)
```

**Function** fuel(selectMDS, distance, Payload)

**Dim** numberStops As Long

numberStops = numStops(selectMDS, distance, Payload)

fuel = fuelCalculations(selectMDS, distance, Payload, numberStops)

**End Function**

## 1.2 Distance

```
'Calculates the vincenty elliptical great circle geodetic curve
```

```
'(Code courtesy of Mike Gavaghan)
```

**Function** vincentyDistance(lat1, long1, lat2, long2) As Double

```
'Ellipsoid properties based on WGS-84
```

**Dim** semiMajor, inverseFlattening, flattening, semiMinor, A, B, big-A, big-B, f, TwoPi, phi1, phi2, lambda, lambda1, lambda2, a2b2b2, tanphi1, tanU1, U1, sinU1 As Double

**Dim** cosU1, tanphi2, tanU2, U2, small\_u2, sinU2, cosU2, sinU1sinU2, cosU1sinU2, sinU1cosU2, cosU1cosU2, a2, b2, omega, i, s As Double

**Dim** sigma, deltasigma, lambda0, sinlambda, coslambda, sin2sigma, sinsigma, cossigma, sinalpha, alpha, cosalpha, cos2alpha, cos2sigmam, cossigmam, cos2sigmam2, C As Double

**Dim** converged As Boolean

**Dim** change As Variant

```
semiMajor = 6378137# 'Meters
```

```

inverseFlattening = 298.257223563
flattening = 1# / inverseFlattening
semiMinor = (1# - flattening) * semiMajor

Application.ScreenUpdating = False

'simplify
A = semiMajor
B = semiMinor
f = flattening
TwoPi = 2# * Application.Pi

'get parameters as radians
phi1 = lat1 * Application.Pi / 180#
lambda1 = long1 * Application.Pi / 180#
phi2 = lat2 * Application.Pi / 180#
lambda2 = long2 * Application.Pi / 180#

'calculations
a2 = A * A
b2 = B * B
a2b2b2 = (a2 - b2) / b2

omega = lambda2 - lambda1

tanphi1 = Tan(phi1)
tanU1 = (1# - f) * tanphi1
U1 = Atn(tanU1)
sinU1 = Sin(U1)
cosU1 = Cos(U1)

tanphi2 = Tan(phi2)
tanU2 = (1# - f) * tanphi2
U2 = Atn(tanU2)
sinU2 = Sin(U2)
cosU2 = Cos(U2)

sinU1sinU2 = sinU1 * sinU2
cosU1sinU2 = cosU1 * sinU2
sinU1cosU2 = sinU1 * cosU2

```

```
cosU1cosU2 = cosU1 * cosU2
```

```
'eq. 13
```

```
lambda = omega
```

```
'intermediates we'll need to compute 's'
```

```
big_A = 0#
```

```
big_B = 0#
```

```
sigma = 0#
```

```
deltastigma = 0#
```

```
converged = False
```

```
For i = 0 To 20
```

```
lambda0 = lambda
```

```
sinlambda = Sin(lambda)
```

```
coslambda = Cos(lambda)
```

```
'eq. 14
```

```
sin2sigma = (cosU2 * sinlambda * cosU2 * sinlambda) + _  
(cosU1sinU2 - sinU1cosU2 * coslambda) ^ 2
```

```
sinsigma = Sqr(sin2sigma)
```

```
'eq. 15
```

```
cossigma = sinU1sinU2 + (cosU1cosU2 * coslambda)
```

```
'eq. 16
```

```
sigma = Application.Atan2(cossigma, sinsigma)
```

```
'eq. 17 Careful! sin2sigma might be almost 0!
```

```
If sin2sigma = 0 Then sinalpha = 0 Else sinalpha = _  
cosU1cosU2 * sinlambda / sinsigma
```

```
Dim check As Variant
```

```
check = sigma * 2
```

```
alpha = Application.Asin(sinalpha)
```

```
cosalpha = Cos(alpha)
```

```
cos2alpha = cosalpha * cosalpha
```

```
'eq. 18 Careful! cos2alpha might be almost 0!
```

```

If cos2alpha = 0# Then cos2sigmam = 0# Else cos2sigmam = -
                                cossigma - 2 * sinU1sinU2 / cos2alpha
small_u2 = cos2alpha * a2b2b2

cos2sigmam2 = cos2sigmam * cos2sigmam

'eq. 3
big_A = 1+small_u2/16384*(4096+small_u2*(-768+small_u2*(320-175*small_u2)))

'eq. 4
big_B = small_u2/1024*(256+small_u2*(-128+small_u2*(74-47*small_u2)))

'eq. 6
deltastigma = big_B*sinsigma*(cos2sigmam+big_B / 4*(cossigma * -
(-1+2*cos2sigmam2)-big_B/6*cos2sigmam*(-3+4*sin2sigma)*(-3+4*cos2sigmam2)))

'eq. 10
C = f / 16 * cos2alpha * (4 + f * (4 - 3 * cos2alpha))

'eq. 11 (modified)
lambda = omega+(1-C)*f*sinalpha*(sigma+C*sinsigma*(cos2sigmam+C*cossigma* -
                                (-1+2*cos2sigmam2)))

'see how much improvement we got
change = Math.Abs((lambda - lambda0) / lambda)

If i > 1 And change < 0.0000000000001 Then
    converged = True
    Exit For
End If
Next

'eq. 19
s = B * big_A * (sigma - deltastigma)

vincentyDistance = s / 1852

End Function

' Given the altitude in 1,000s of feet , Klbs reserve , alternate and holding (rah) fuel ,

```

```

'Klbs payload , Klbs fuel consumed (max fuel after payload) in Klbs ,
'determines the max distance possible
Function maxDistance(selectMDS , Payload) As Double
  Dim wop , wgmt , wmaxfuel , wpmmax , wfrah , wfstto , wfapp , alt , A , B , C , G , -
  totalFuelConsumed As Double
  Dim cruise , climb , descent , given As Range

  Set cruise = Range("cruiseBeta")
  Set climb = Range("climbBeta")
  Set descent = Range("descentBeta")
  Set given = Range("GivensTable")

  maxDistance = 0
  plane = getPlane(selectMDS)

  'for ease of reading the equations .
  wop = given(plane , 2)
  wgmt = given(plane , 3)
  wmaxfuel = given(plane , 4)
  wpmmax = given(plane , 5)
  wfrah = given(plane , 6)
  wfstto = given(plane , 7)
  wfapp = given(plane , 8)
  alt = given(plane , 9)

  'difference from max gross takeoff and payload and reserve
  totalFuelConsumed = wgmt - wop - Payload - wfrah

  If totalFuelConsumed+wfrah > wmaxfuel Then totalFuelConsumed = wmaxfuel-wfrah

  'Use Formula distance = A*g^3 + B*g^2 + C*g coefficients . G represents fuel consumed .
  A = cruise(plane , 6)/3
  B = cruise(plane , 5)/2+cruise(plane , 6)*(wop+Payload+wfstto+wfapp+wfrah)+ -
  cruise(plane , 7) * (alt) / 2# '( _3 /2+ _4 *(EW+w)+ _5 /2* Alt)
  C = cruise(plane , 2) + cruise(plane , 3) * alt + cruise(plane , 4) * (alt) ^ 2 + -
  cruise(plane , 5) * (wop + Payload + wfstto + wfapp + wfrah) + -
  cruise(plane , 6) * (wop + Payload + wfstto + wfapp + wfrah) ^ 2 + -
  cruise(plane , 7) * alt * (wop + Payload + wfstto + wfapp + wfrah)
  '( _0 + _1 *Alt+ _2 *A?lt?^2+ _3 *(EW+w)+ _4 *(EW+w)^2+ _5 *Alt*(EW+w))
  G = totalFuelConsumed

```

```

    maxDistance = A * ((G ^ 3) + B * ((G ^ 2) + C * G
End Function

'calculates the number of stops required for an MDS given the distance and payload
Function numStops(selectMDS, distance, Payload) As Double
    Dim stopDist As Double

    stopDist = maxDistance(selectMDS, Payload)
    numStops = distance / stopDist
    numStops = Application.WorksheetFunction.RoundDown(numStops, 0)
End Function

'given source and destination ICAOs this function finds the associated lats and longs
'and returns the distance between them
Function myDistance(fromString, toString) As Double

    Dim rowCount, colCount, aircraftCount, totalFromBases, totalToBases, totalNumAircraft, _
    i, k, col As Long
    Dim tempstring, fromPCN, toPCN As String
    Dim lat1, lat2, long1, long2, Payload, distance, baseFuel, maxpayload As Double
    Dim endCond As Boolean

    'Find lat/long data for "from base
    For i = 1 To 5342
        If Sheets("AirfieldData").Cells(i, 2) = fromString Then
            lat1 = Sheets("AirfieldData").Cells(i, 3)
            long1 = Sheets("AirfieldData").Cells(i, 4)
            'fromPCN = Sheets("AirfieldData").Cells(i, 7)
        Exit For
    End If
Next

    'Find the lat/long for the "to" base
    For i = 1 To 5342
        If Sheets("AirfieldData").Cells(i, 2) = toString Then
            lat2 = Sheets("AirfieldData").Cells(i, 3)
            long2 = Sheets("AirfieldData").Cells(i, 4)
            'toPCN = Sheets("AirfieldData").Cells(i, 7)
        Exit For

```

```

        End If
    Next i

    'Populate distance
    myDistance = vincentyDistance(lat1, long1, lat2, long2)

```

**End Function**

### 1.3 Pre-Processing

```

'this sub moves through all the source and destination combinations and populates
'the Aircraft sheet with the distance between them, fuel cost per payload pound,
'and basic fuel cost given the aircraft MDS

```

**Sub** getDistance()

```

    Dim rowCount, colCount, aircraftCount, totalFromBases, totalToBases, _
    totalNumAircraft, i, k, col As Long

```

```

    Dim fromString, toString, tempstring, fromPCN, toPCN As String

```

```

    Dim lat1, lat2, long1, long2, Payload, distance, baseFuel, maxpayload, _
    payloadFuel As Double

```

```

    Dim endCond As Boolean

```

```

    Application.ScreenUpdating = False

```

```

    rowCount = 0

```

```

    colCount = 0

```

```

    aircraftCount = 0

```

```

    Payload = 0

```

```

Do While endCond = False

```

```

    tempstring = Sheets("SupplyBases").Cells(rowCount + 2, 1)

```

```

If IsEmpty(tempstring) Then

```

```

    endCond = True

```

```

    Exit Do

```

```

End If

```

```

    rowCount = rowCount + 1

```

**Loop**

```

    totalFromBases = rowCount

```

```

    endCond = False

```

```

Do While endCond = False

```

```

    tempstring = Sheets("DemandBases").Cells(colCount + 2, 1)

```

```

If IsEmpty(tempstring) Or tempstring = "" Then
    endCond = True
    Exit Do
End If
    colCount = colCount + 1
Loop

totalToBases = colCount

endCond = False
Do While endCond = False
    tempstring = Sheets("Aircraft").Cells(aircraftCount + 8, 1)
    If IsEmpty(tempstring) Or tempstring = "" Then
        endCond = True
        Exit Do
    End If
    aircraftCount = aircraftCount + 1
Loop

totalNumAircraft = aircraftCount
col = 1

Application.ScreenUpdating = False
For rowCount = 1 To totalFromBases
    fromString = Sheets("SupplyBases").Cells(rowCount + 1, 1)
    'Find lat/long data for "from base"
    For i = 1 To 5342
        If Sheets("AirfieldData").Cells(i, 2) = fromString Then
            lat1 = Sheets("AirfieldData").Cells(i, 3)
            long1 = Sheets("AirfieldData").Cells(i, 4)
            fromPCN = Sheets("AirfieldData").Cells(i, 7)
            Exit For
        End If
    Next

    'Find the lat/long for the "to" base
    For colCount = 1 To totalToBases
        toString = Sheets("DemandBases").Cells(colCount + 1, 1)
        If Not toString = fromString Then
            For i = 1 To 5342

```

```

If Sheets(" AirfieldData").Cells(i, 2) = toString Then
    lat2 = Sheets(" AirfieldData").Cells(i, 3)
    long2 = Sheets(" AirfieldData").Cells(i, 4)
    toPCN = Sheets(" AirfieldData").Cells(i, 7)
    Exit For
End If
Sheets(" Aircraft").Cells(6, 14 + col) = fromString
Sheets(" Aircraft").Cells(7, 14 + col) = toString
Sheets(" maxfuel").Cells(1, 1 + col) = fromString
Sheets(" maxfuel").Cells(2, 1 + col) = toString
Sheets(" basefuel").Cells(1, 1 + col) = fromString
Sheets(" basefuel").Cells(2, 1 + col) = toString

```

**Next i**

```

'Populate distance and fuel cost/(payload Klb) given aircraft type for all
'aircraft on " aircraft" sheet

```

```

distance = vincentyDistance(lat1, long1, lat2, long2)

```

**For k = 1 To totalNumAircraft**

```

selectMDS = Sheets(" Aircraft").Cells(k + 7, 2).Value
Payload = Sheets(" Aircraft").Cells(k + 7, 6).Value
maxpayload = Sheets(" Aircraft").Cells(k + 7, 6).Value

```

```

Sheets(" Aircraft").Cells(k+7, 14).Value = -
    Sheets(" Aircraft").Cells(k+7, 1)
Sheets(" maxfuel").Cells(k+2, 1).Value = -
    Sheets(" Aircraft").Cells(k+7, 1)
Sheets(" basefuel").Cells(k+2, 1).Value = -
    Sheets(" Aircraft").Cells(k+7, 1)

```

```

Sheets(" Aircraft").Cells(k + 7, 13 + col + 1).Value = distance
'Sheets(" Aircraft").Cells(k + 3, 14 + col).Value = -
airfieldNodalReduction(selectMDS, lat1, long1, lat2, long2, -
fromPCN, toPCN, payload)
baseFuel = fuel(selectMDS, distance, 0)
payloadFuel = fuel(selectMDS, distance, Payload)

```

```

Sheets(" maxfuel").Cells(k + 2, 1 + col).Value = -
(payloadFuel - baseFuel) / maxpayload

```

```

        Sheets("basefuel").Cells(k + 2, 1 + col).Value = baseFuel
    Next k
    col = col + 1
End If
Next colCount

Next rowCount
Application.ScreenUpdating = True
End Sub

```

## 1.4 Post-Processing

```

Sub FinalCost()
    Dim endCond As Boolean
    Dim tempstring, toString, fromString, planeString As String
    Dim aircraftCount, allocatedCount, baseCount, i, j, k As Long
    Dim distance, Payload As Double

    aircraftCount = 0
    allocatedCount = 0
    baseCount = 0

    Application.ScreenUpdating = False

    'count the number of aircraft
    endCond = False
    Do While endCond = False
        tempstring = Sheets("Aircraft").Cells(aircraftCount + 8, 1)
        If IsEmpty(tempstring) Or tempstring = "" Then
            endCond = True
            Exit Do
        End If
        aircraftCount = aircraftCount + 1
    Loop

    'count the number of allocated aircraft
    endCond = False
    Do While endCond = False
        tempstring = Sheets("ACAllocation").Cells(1, allocatedCount + 3)
        If IsEmpty(tempstring) Or tempstring = "" Then

```

```

        endCond = True
    Exit Do
End If
    allocatedCount = allocatedCount + 1
Loop

'count the number of bases
endCond = False
Do While endCond = False
    tempstring = Sheets("ACAllocation").Cells(baseCount + 2, 1)
If IsEmpty(tempstring) Or tempstring = "" Then
        endCond = True
    Exit Do
End If
    baseCount = baseCount + 1
Loop

For i = 1 + 1 To baseCount + 1 'go down the rows of bases
    fromString = Sheets("ACAllocation").Cells(i, 1)
    toString = Sheets("ACAllocation").Cells(i, 2)
For j = 1 + 2 To allocatedCount + 3 'go across the columns of allocated
    planeString = Sheets("ACAllocation").Cells(1, j)
For k = 1 To aircraftCount 'loop through the total aircraft
    tempstring = Sheets("Aircraft").Cells(k + 7, 1)
If planeString = tempstring And -
        Sheets("ACAllocation").Cells(i, j) > 0 Then
            selectMDS = Sheets("Aircraft").Cells(k + 7, 2).Value
            distance = myDistance(fromString, toString)
            Payload = Sheets("FinalPayload").Cells(i, j).Value
            Sheets("finalFuel").Cells(1, j) = planeString
            Sheets("finalFuel").Cells(i, 1) = fromString
            Sheets("finalFuel").Cells(i, 2) = toString
            'cycle cost is payload to destination, empty back
            Sheets("finalFuel").Cells(i, j) = -
                fuel(selectMDS, distance, Payload)+fuel(selectMDS, distance, 0)
Exit For
End If
Next k
Next j
Next i

```

```
Application.ScreenUpdating = True  
End Sub
```

## Appendix B. General Algebraic Modeling System Code

### Sets

```
i supply bases
j demand bases
k item types /super, standard/
l aircraft
c index for climb /0*8/
ws given weights;
```

### Parameters

```
Ed(j,k)      expected demand at base j for item k
s(i,k)       supply inventory at base i of item k
w(k)         weight of item k
h(j,k)       holding cost ($) at base j of item k
p(j,k)       penalty for shorting demand at base j of item k
alt(l)       the altitude of aircraft l
wg(l,ws)     weight parameters of aircraft l

maxFuel(l,i,j) fuel to fly the max payload for distance i j and acrft
baseFuel(l,i,j);
```

```
$onecho > taskout.txt
```

```
Set = i          rng=SupplyBases!a1  rdim=1
Set = j          rng=DemandBases!a1  rdim=1
Set= l          rng=Aircraft!a8      rdim=1
Set=ws         rng=Aircraft!b7:K7      cdim=1 IgnoreColumns=B
Par=w          rng=Items!b2          rdim=1
Par = wg       rng=Aircraft!a7      rdim=1 cdim=1 IgnoreColumns=B
Par= maxFuel   rng=maxfuel!a1       rdim=1 cdim=2
Par=baseFuel   rng=basefuel!a1      rdim=1 cdim=2
Par=Ed         rng=DemandBases!a1   rdim=1 cdim=1
$offecho
```

```
$call GDXXRW indata.xlsm trace=3 @taskout.txt
```

```
$GDXIN indata.gdx
```

```
$LOADDC i j l ws wg maxFuel baseFuel Ed
```

```
$GDXIN
```

### Variables

```

z          objective function value;

z.lo=0;
z.up=100000;

Positive Variables
wp(i,j,l)   payload weight for route i to j on aircraft l
x(i,j,k,l)  number of pallets from i to j of item k to fly on aircraft l ;
wp.up(i,j,l)=270; #maximum payload weight of C5
x.up(i,j,k,l)=36;

Binary Variables
y(i,j,k,l)  1 if from base i to base j aircraft l is assigned to deliver something

Equations
objfun          defines the overall objective function
meetexpdemand(j,k)  Must meet expected demand
payloadWeightbound(i,j,k,l) cannot exceed aircraft weight capacity
spacebound(i,j,k,l)   cannot exceed aircraft floor space capacity
oneassignmentperac(l)  only assign an aircraft to one i-j combination
* supply(i,k)         cannot exceed available supply
payload(i,j,l)       payload weight
assignmentperK(i,j,l)  only assign one K-type per aircraft allocation
;

objfun .. z=e= 1*sum(k,sum(i,sum(j,sum(l,(maxFuel(l,i,j)*wp(i,j,l)+
                                baseFuel(l,i,j))*y(i,j,k,l)))));

*****Constraints*****
meetexpdemand(j,k)..      sum(i,sum(l,x(i,j,k,l))) =e= Ed(j,k);
payloadWeightbound(i,j,k,l)..  w(k)*x(i,j,k,l) =l= wg(l,'wapmax')*y(i,j,k,l);
spacebound(i,j,k,l)..      x(i,j,k,l)=l= wg(l,'scap')* y(i,j,k,l);
oneassignmentperac(l)..    sum(k,sum(i,sum(j,y(i,j,k,l)))) =l= 1;
payload(i,j,l)..          wp(i,j,l)=e=w('super')*x(i,j,'super',l)+
                                w('standard')*x(i,j,'standard',l) ;
assignmentperK(i,j,l)..    sum(k,y(i,j,k,l))=l=1;

Model MausModel / all /;

```

```

*Ed('ETAR','super')=50; Ed('ETAR','standard')=13;
*Ed('RKSO','super')=50; Ed('RKSO','standard')=13;

w('super')=2.6; w('standard')=2.6;

option nlp=minos;
option minlp=baron;

Solve MausModel minimizing z using minlp;

Display y.l;
Display x.l;
Display wp.l;

execute_unload 'indata.gdx', x,y,wp, z, wg;
execute 'gdxxrw.exe indata.gdx O=indata.xlsxm var=x.l rng=CargoAllocation!a1' ;
execute 'gdxxrw.exe indata.gdx O=indata.xlsxm var=y.l rng=ACAllocation!a1' ;
execute 'gdxxrw.exe indata.gdx O=indata.xlsxm var=wp.l rng=FinalPayload!a1' ;;
execute 'gdxxrw.exe indata.gdx O=indata.xlsxm var=z.l rng=GAMScost!a1' ;

```

## Bibliography

1. T. Lipp, “BARON : Branch And Reduce Optimization Navigator (Slide Deck for Stanford University CME334).”  
<https://web.stanford.edu/class/cme334/docs/2011-12-01-BARON>, 2011.
2. A. D. Reiman, *Enterprise Analysis of Strategic Airlift to Obtain Competitive Advantage through Fuel Efficiency*. AFIT-ENS-DS-14-S-16. School of Engineering and Management, Air Force Institute of Technology (AU), Wright Patterson AFB OH, 2014.
3. S. M. Boone, “Shortest Path Across Stochastic Network with Correlated Random Arcs.” MS thesis, MS thesis, AFIT-ENC-MS-18-M-109. School of Engineering and Management, Air Force Institute of Technology (AU), Wright Patterson AFB OH, 2018.
4. J. D. Maywald, A. D. Reiman, A. W. Johnson, and R. E. Overstreet, “The Myth of Strategic and Tactical Airlift,” *Air and Space Power Journal*, vol. 31, no. 1, pp. 61–71, 2017.
5. H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Berlin: Springer-Verlag, 1 ed., 2010.
6. R. M. Karp, “Reducibility Among Combinatorial Problems,” in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pp. 219–242, New York City: Springer, 2010.
7. C. Wilbaut, S. Hanafi, and S. Salhi, “A Survey of Effective Heuristics and their Application to a Variety of Knapsack Problems,” *IMA Journal of Management Mathematics*, vol. 19, no. 3, pp. 227–244, 2008.
8. S. Kosuch and A. Lisser, “On Two-Stage Stochastic Knapsack Problems,” *Discrete Applied Mathematics*, vol. 159, no. 16, pp. 1827–1841, 2011.
9. T. C. Perry and J. C. Hartman, “An Approximate Dynamic Programming Approach to Solving a Dynamic, Stochastic Multiple Knapsack Problem,” *International Transactions in Operational Research*, vol. 16, no. 3, pp. 347–359, 2009.
10. K. W. Ross and D. H. K. Tsang, “The Stochastic Knapsack Problem,” *IEEE Transactions on Communications*, vol. 37, no. 7, pp. 740–747, 1989.
11. S. Martello and P. Toth, “0-1 Multiple Knapsack Problem,” in *Knapsack Problems*, ch. 6, pp. 157–187, Chichester: John Wiley & Sons Ltd., 1990.
12. N. C. Samuel Eilon, “The Loading Problem,” *Management Science*, vol. 51, no. 7, pp. 1092–1100, 1971.

13. X. Lai, J. K. Hao, F. Glover, and Z. Lü, "A Two-Phase Tabu-Evolutionary Algorithm for the 01 Multidimensional Knapsack Problem," *Information Sciences*, vol. 436-437, pp. 282–301, 2018.
14. J. Simon, A. Apte, and E. Regnier, "An Application of the Multiple Knapsack Problem: The Self-Sufficient Marine," *European Journal of Operational Research*, vol. 256, no. 3, pp. 868–876, 2017.
15. W. Shih, "A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem," *Journal of the Operational Research Society*, vol. 30, no. 4, pp. 369–378, 1979.
16. E. Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research*, vol. 13, no. 4, pp. 517–545, 1965.
17. M. Vasquez and J. K. Hao, "A Hybrid Approach for the 0-1 Multi Knapsack Problem," in *Proceedings of the International Joint Conference on Artificial Intelligence*, (Seattle, Washington), pp. 328–333, 2001.
18. B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, "A Hybrid Quantum Particle Swarm Optimization for the Multidimensional Knapsack Problem," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 1–13, 2016.
19. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. New Jersey: Prentice-Hall, Inc., 1 ed., 1993.
20. W. L. Winston and J. B. Goldberg, *Operations Research: Applications and Algorithms*. Toronto: Thomson Brooks/Cole, 4 ed., 2004.
21. R. Vaswani, "The Aircraft Assignment Problem," *Aeronautical Engineering Review*, vol. 16, no. 2, pp. 64–68, 1957.
22. A. R. Ferguson and G. B. Dantzig, "The Problem of Routing Aircraft," *Aeronautical Engineering Review*, vol. 14, no. 4, pp. 51–55, 1955.
23. A. R. Ferguson and G. B. Dantzig, "The Allocation of Aircraft to Routes - An Example of Linear Programming Under Uncertain Demand," *Management Science*, vol. 3, no. 1, pp. 45–73, 1956.
24. D. T. Wu and S. M. Ross, "A Stochastic Assignment Problem," *Naval Research Logistics*, vol. 62, no. 1, pp. 23–31, 2015.
25. S. Kataoka and T. Yamada, "Upper and Lower Bounding Procedures for the Multiple Knapsack Assignment Problem," *European Journal of Operational Research*, vol. 237, no. 2, pp. 440–447, 2014.

26. N. B. Dimitrov, D. Solow, J. Szmerekovsky, and J. Guo, "Emergency Relocation of Items using Single Trips: Special cases of the Multiple Knapsack Assignment Problem," *European Journal of Operational Research*, vol. 258, no. 3, pp. 938–942, 2017.
27. W. Cook, "Markowitz and Manne + Eastman + Land and Doig = Branch and Bound," *Documenta Mathematica*, vol. 17, no. 2, pp. 227–238, 2012.
28. A. H. Land and A. G. Doig, "An Automatic Method for Solving Discrete Programming Problems," in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pp. 105–106, Berlin: Springer-Verlag, 2010.
29. J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem," *Operations Research*, vol. 11, no. 6, pp. 972–989, 1963.
30. N. Sahinidis, "BARON User Manual v. 2018.11.15." <https://minlp.com/baron-downloads>, 2018.
31. M. Tawarmalani and N. Sahinidis, "Global Optimization of Mixed-Integer Nonlinear Programs : a Theoretical and Computational Study," *Mathematical Programming*, vol. 99, no. 3, pp. 563–591, 2004.
32. "GAMS The Solver Manuals." Gams Development Corporation. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.8560&rep=rep1&type=pdf>, 2008.
33. R. K. Nangia, "Operations and Aircraft Design Towards Greener Civil Aviation using Air-to-Air Refuelling," *The Aeronautical Journal*, vol. 110, no. 1113, pp. 705–721, 2006.
34. S. Baker, D. Morton, R. Rosenthal, and L. Williams, "Optimizing Military Airlift," *Operations Research*, vol. 50, no. 4, pp. 582–602, 2002.
35. T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations," *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975.
36. "World Geodetic System 1984 (WGS 84)." National Geospatial-Intelligence Agency. Retrieved on 18 December 2018. <http://earth-info.nga.mil/GandG/update/index.php?dir=wgs84action=wgs84>.
37. "C-5 A / B / C Galaxy and C-5M Super Galaxy." Department of Defense, 2017. Retrieved on 24 October 2018. <https://www.amc.af.mil/About-Us/Fact-Sheets/Display/Article/977534/c-5-abc-galaxy-and-c-5m-super-galaxy/>.

38. "C-17 Globemaster III." Department of Defense, 2018. Retrieved on 13 November 2018. <https://www.af.mil/About-Us/Fact-Sheets/Display/Article/1529726/c-17-globemaster-iii/>.
39. "C-130 Hercules." Department of Defense, 2018. Retrieved on 13 November 2018. <https://www.af.mil/About-Us/Fact-Sheets/Display/Article/1555054/c-130-hercules/>.
40. M. R. Kılıç and N. V. Sahinidis, "Exploiting Integrality in the Global Optimization of Mixed-Integer Nonlinear Programming Problems with BARON," *Optimization Methods and Software*, vol. 33, no. 3, pp. 540–562, 2018.
41. "Channel Performance and Analysis Report (Cargo)." 618th Air Operations Center Tanker Airlift Control Center, 2018.
42. "Channel Performance and Analysis Report (Cargo)." 618th Air Operations Center Tanker Airlift Control Center, 2019.

# 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.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 21-03-2019		<b>2. REPORT TYPE</b> Master's Thesis		<b>3. DATES COVERED (From — To)</b> Sept 2017 — Mar 2019	
<b>4. TITLE AND SUBTITLE</b>  Applying the Multiple Multidimensional Knapsack Assignment Problem to a Cargo Allocation and Transportation Problem with Stochastic Demand				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
<b>6. AUTHOR(S)</b>  Maus, Jocelin S., Captain, USAF				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  AFIT-ENS-MS-19-M-137	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  Intentionally Left Blank				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b>  DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
<b>13. SUPPLEMENTARY NOTES</b>  This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
<b>14. ABSTRACT</b>  Utilizing fiscal year 2017-2018 cargo data published by the 618th Air Operations Center and modeling this problem as a multiple multidimensional knapsack assignment problem (MMKAP), this work investigates how categorical assumptions about demand affect aircraft allocation and assesses the economic penalties associated with shorting or exceeding demand in the event of mis-estimation given a stochastic demand. This work starts with the general formulation of a new variant of the MMKAP and applies the MMKAP to a notional military airlift example with two supply, two demand nodes, two item types, and three aircraft types. After a deterministic solution is found, the effects of a stochastic demand are explored using different cost models and random draws from distribution functions based on reported cargo shipment data. This research concludes that there are levels at which demand expectations can be set to mitigate economic penalties given a fixed cost penalty and a variable cost penalty.					
<b>15. SUBJECT TERMS</b>  Multiple Multidimensional Knapsack Assignment Problem, MMKAP, Military Airlift, Stochastic Demand					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>	<b>b. ABSTRACT</b>	<b>c. THIS PAGE</b>			Maj Thomas P. Talafuse, AFIT/ENS
U	U	U	UU	78	<b>19b. TELEPHONE NUMBER (include area code)</b> (937)255-3636x4740; Thomas.Talafuse@afit.edu