A MULTI-PASS CONSTRUCTION HEURISTIC FOR THE AGGREGATED AIRLIFT PROBLEM

THESIS

Stephen T. O’Leary, Captain, USAF

AFIT/GOR/ENS/06-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.
A MULTI-PASS CONSTRUCTION HEURISTIC FOR THE AGGREGATED AIRLIFT PROBLEM

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

Stephen T. O’Leary, BS
Captain, USAF

March 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
A MULTI-PASS CONSTRUCTION HEURISTIC
FOR THE AGGREGATED AIRLIFT PROBLEM

Stephen T. O’Leary, BS
Captain, USAF

Approved:

Dr. Gary W. Kinney, Maj, USAF (Chairman)  date

Dr. James T. Moore (Member)  date
Abstract

The Air Force has over 26,000 Airmen deployed from over 80 military installations to Forward Operating Locations (FOLs) in over 21 countries providing capabilities for air mobility and refueling, precision strike, close air support, intelligence, surveillance, and reconnaissance among others. Airmen are rotated in and out of these deployed positions every 120 days as part of the AEF system. Prior to January 2005, small groups of deploying personnel made their way via ad hoc methods providing little to no visibility to planners and creating a backlog of personnel at the Points of Debarkation (PODs) where deploying airmen often waited up to 5 days before receiving tactical airlift to their FOLs. The Single Ticket process was designed to alleviate this situation by scheduling transportation for airmen from their homebase to their FOL. In-Transit Visibility (ITV) is required to properly implement the Single Ticket process. Proper ITV is provided by aggregated airlift. With aggregated airlift personnel are aggregated into groups of 100 or more based on region of origin, POD and Required Delivery Date (RDD) so that dedicated airlift can be generated for their deployment. The dedicated airlift facilitates better ITV of personnel moving through the PODs, enables proper planning and scheduling of intratheater airlift from the POD to the FOL, and reduces the backlog of personnel and wait times at the POD. In the aggregated airlift problem we are given a Time Phased Force Deployment Document (TPFDD) containing a list of personnel scheduled for deployment, their base of origin, POD, FOL, and RDD. This research creates a multi-pass construction heuristic to solve the aggregated airlift problem. The algorithm was tested on several notional and real world TPFDDs and performed well, increasing the effectiveness over the manual process by more than 20%.
Dedication

To My Wife, who kept everything running perfectly and to my two sons, of whom I am so proud; I could not have done this without you! I love you all and appreciate the sacrifices you made throughout my course of study.
Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Maj Gary Kinney, for his guidance and support throughout the course of this thesis effort. His insight and expert knowledge of heuristics was certainly appreciated. I would, also, like to thank my sponsor, BGen Stephen L. Hoog, Commander, AEFC, for giving me the opportunity to research and help solve this problem.

I am, also, indebted to the Mr. Mitchell Berry, an analyst at AEFC, who was an excellent point of contact for all my questions on the aggregation process.

Stephen T. O’Leary
# Table of Contents

Abstract .............................................................................................................................. iv
Dedication ........................................................................................................................... v
Acknowledgments .............................................................................................................. vi
Table of Contents .............................................................................................................. vii
List of Tables ................................................................................................................... viii
List of Figures ................................................................................................................... ix
List of Acronyms ................................................................................................................ x
Introduction: ........................................................................................................................ 1
  1.1 Overview ..................................................................................................................... 1
  1.2 Single Ticket Deployments ....................................................................................... 3
  1.3 Problem Statement: ................................................................................................. 4
  1.4 Document Outline ................................................................................................. 4
2. Background ..................................................................................................................... 5
  2.1 The AEF Concept ..................................................................................................... 5
  2.2 Air Force Deployments ........................................................................................... 5
  2.3 The Single Ticket Deployment Process ................................................................... 9
  2.4 Aggregated Airlift ................................................................................................... 10
  2.5 Airlift Aggregation as an Integer Program ............................................................. 11
  2.6 The Hub Allocation Problem .................................................................................. 14
  2.7 Heuristic Search Methods ..................................................................................... 15
3. Methodology .................................................................................................................. 18
  3.1 Overview .................................................................................................................. 18
  3.2 Identifying Extraneous RDDs and Creating a Bound .............................................. 18
  3.3 Generating a Starting Solution ............................................................................... 19
  3.5 Combining Smaller Aggregates .......................................................................... 22
  3.6 Combining ULNs in Infeasible Aggregates ........................................................... 31
  3.7 Output and Post Processing ................................................................................... 31
4. Testing and Results ....................................................................................................... 35
  4.1 Introduction .............................................................................................................. 35
  4.2 Testing for performance on Notional TPFDDs ....................................................... 35
  4.3 Testing for Performance on Real World TPFDDs ............................................... 38
  4.4 Analysis of Removing the Regions Constraint ...................................................... 40
5. Conclusion .................................................................................................................... 45
  5.1 Summary of Results ............................................................................................ 45
  5.2 Airlift Aggregation for AEF Cycle 1/2 ................................................................. 45
  5.3 Further Research ................................................................................................. 46
Bibliography ...................................................................................................................... 47
Appendix A: VBA Code ..................................................................................................... 49
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1.</td>
<td>Notional TPFDD</td>
<td>20</td>
</tr>
<tr>
<td>Table 2.</td>
<td>Sample Output</td>
<td>33</td>
</tr>
<tr>
<td>Table 3.</td>
<td>Aggregation Results for Notional TPFDDs</td>
<td>37</td>
</tr>
<tr>
<td>Table 4.</td>
<td>Confidence Interval</td>
<td>38</td>
</tr>
<tr>
<td>Table 5.</td>
<td>Aggregation Results for Real World TPFDDs</td>
<td>39</td>
</tr>
<tr>
<td>Table 6.</td>
<td>Aggregation Results of Notional TPFDDs with Region Constraint Relaxed</td>
<td>41</td>
</tr>
<tr>
<td>Table 7.</td>
<td>Aggregation Results of Real World TPFDDs with Region Constraint Relaxed</td>
<td>42</td>
</tr>
<tr>
<td>Table 8.</td>
<td>Sample Output With and Without Region Constraint</td>
<td>43</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Sample ULN Data from JOPES</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Aggregate Possibilities for a Particular RDD</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Phase 1 Combination</td>
<td>24</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Phase 2 Combination</td>
<td>25</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Phase 3 Combination</td>
<td>26</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Phase 4 Combination</td>
<td>27</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Phase 5 Combination</td>
<td>28</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Phase 6 Combination</td>
<td>29</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Phase 7 Combination</td>
<td>30</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Passenger Distributions of Real World TPFDDs</td>
<td>40</td>
</tr>
</tbody>
</table>
List of Acronyms

AEF ............................................................... Air & Space Expeditionary Force
AEFC ........................................................... Air & Space Expeditionary Force Center
AETF ............................................................. Air & Space Expeditionary Task Force
AMC ........................................................................................................ Air Mobility Command
BWI ........................................................................................................... Baltimore Washington International Airport
EAD ........................................................................................................... Earliest Arrival Date
FOL ........................................................................................................ Forward Operating Location
IDO ........................................................................................................... Installation Deployment Officer
ITV ........................................................................................................... In Transit Visibility
JOPES ................................................................. Joint Planning and Execution System
LAD ........................................................................................................... Latest Arrival Date
PID ........................................................................................................... Plan ID
POD ........................................................................................................... Point of Debarkation
POE ........................................................................................................... Point of Embarkation
PRU ........................................................................................................ Personnel Readiness Unit
RDD ........................................................................................................... Required Delivery Date
RLD ........................................................................................................... Ready to Load Date
TACC ........................................................................................................ Tanker Airlift Control Center
TMO ........................................................................................................... Traffic Management Office
TPFDD ............................................................. Time Phased Force Deployment Document (Data)
TPFDDL ........................................................ Time Phased Force Deployment Data List
UDC ............................................................................................................................ Unit Deployment Center
ULN .................................................................................................................................... Unit Line Number
UTC .................................................................................................................................... Unit Type Code
A MULTI-PASS CONSTRUCTION HEURISTIC
FOR THE AGGREGATED AIRLIFT PROBLEM

Introduction:

1.1 Overview

The Air Force currently has more than 26,000 Airmen deployed in support of the Global War on Terror and Combatant Commanders. These personnel are deployed from over 80 military installations to 41 Forward Operating Locations (FOLs) in 21 different countries. They support over 300 aircraft providing capabilities for air mobility and refueling, precision strike, close air support, intelligence, surveillance, and reconnaissance among others.

To keep all of these positions filled the Air Force uses the Air & Space Expeditionary Force (AEF) concept. Under this concept the Air Force can rotate Airmen in and out of theater to support long term operations while maintaining a high level of training and expertise. Each airman is vulnerable for deployment for 4 months of a 20-month cycle which allows predictability of deployment for all Airmen and their units.

The Air & Space Expeditionary Force Center (AEFC) executes the Air Force battle rhythm and is the primary functional unit for sourcing requirements for the deployment of the United States Air Force. AEFC is responsible for filling each of the over 26,000 positions for each AEF cycle with personnel currently stationed all over the world (Herbert, 2003). AEFC works closely with Air Mobility Command (AMC) to
deploy Air Force personnel and equipment giving the Air Force the capability to rotate personnel in and out the deployed positions every 120 days.

The military deploys by Unit Type Code (UTC), a designator including everything from a large fighter squadron and all of its support personnel and equipment to a single Contracting Officer with only hand carried baggage. When deploying groups of 100 or more from the same wing, units process their personnel and cargo through the Unit Deployment Center (UDC) led by the Installation Deployment Officer (IDO). After processing, dedicated cargo and passenger planes pick them up from their Point of Embarkation (POE) and take them overseas.

Prior to January 2005, smaller deployments of less than 100 people were treated differently. Members processed through their Personnel Readiness Unit (PRU) and cargo was sent through the Traffic Management Office (TMO). Personnel flew commercial air to Baltimore Washington International Airport, boarded a “Patriot Express” contracted airlift mission to their Point of Debarkation (POD) and awaited intratheater airlift to their final destination. (AFI 10-403). Commercial airlift and “Patriot Express” missions provide planners at PODs with little to no In-Transit Visibility (ITV) or advanced notice of incoming airmen. This lack of visibility made it impossible to schedule adequate intratheater airlift to transport personnel to their FOLs. This resulted in up to a 5 day backlog at the PODs and unnecessary delays for Airmen to reach their FOLs (Hoog, 2005).
1.2 Single Ticket Deployments

Currently, AEFC is using the “Single Ticket” system on all deployments to schedule airlift for airmen from their home station all the way to their FOL (Hoog, 2005). A passenger’s travel itinerary is dictated in the Time Phased Force Deployment Document (TPFDD) from their POE to their final destination or FOL (USCENTAF, 2005). The process of assigning airlift to all passengers is very time consuming and can delay publishing and updating the TPFDD.

To schedule follow-on aircraft, an accurate forecast of incoming passengers is required. Mission planning and ITV are facilitated by using dedicated airlift to move personnel. The minimum number of passengers for dedicated airlift is 100. Two groups of 40 or more (totaling more than 100) can be collected from two POEs in CONUS and two groups of 40 or more (totaling more than 100) can be delivered to two PODs in theater. This is the purpose of aggregated airlift – gathering UTCs within a region and a common Required Delivery Date (RDD) window so dedicated airlift from POE to POD can be requested. Aggregated airlift was used for AEF 7/8, AEF 9/10 and is planned for AEF 1/2. This research creates a mathematical algorithm to partially automate airlift aggregation. The solution will allow more thorough travel planning and efficient deployment of personnel. The overall goal is to shorten the backlog at PODs to less than 24 hours by giving planners at PODs an accurate forecast of passengers transiting through their ports.
1.3 Problem Statement:

The problem is given a sourced TPFDD with origins, PODs, destinations, and RDDs; place all personnel in groups of at least 100 traveling from the same region to the same POD and having RDDs within a 5 day span for Active Duty passengers and a 3 day span for Reserve/Guard personnel. A group can be composed of personnel from two regions if there are at least 40 from each region. Similarly, a group can travel to two PODs if there are at least 40 personnel traveling to each POD. If possible, passengers from the same origin with the same RDDs should be in the same group, and the span of the RDD’s should be minimized. The overall objective is to minimize the number of personnel not in groups.

1.4 Document Outline

The outline of the remainder of this document is as follows. Chapter 2 provides a background on the AEF concept and Air Force deployment process, a survey of similar problems in logistics, and an introduction to heuristic problem solving. The methodology of the algorithm used to automate airlift aggregation is discussed in Chapter 3. In Chapter 4, the algorithm is tested on a set of randomly generated TPFDDs and two real world TPFDDs from AEF 7/8 and AEF 9/10. Finally in Chapter 5, summarizes the results and proposes further research opportunities pertaining to this problem.
2. Background

“The Air & Space Expeditionary Force (AEF) is the USAF methodology for organizing, training, equipping, and sustaining rapidly responsive air and space forces to meet the defense strategy requirements. Through the AEF, the Air Force supports defense strategy requirements using a combination of both permanently assigned and rotational forces. The USAF organizes Air & Space Expeditionary Task Forces (AETF) to meet defense strategy requirements using both CONUS based and forward stationed units.” – Air Force Planning Document 10-4

2.1 The AEF Concept

To adequately sustain the force and forecast deployments for Airmen, the Air Force deploys personnel on a 120-day rotation schedule as part of an Air & Space Expeditionary Task Force. AEF is the methodology or system under which the Air Force deploys. The AEF concept allows the Air Force to better schedule units for deployments and affords most Airmen predictability in their deployments. Each airman is aligned to one AEF scheduled on a 20-month rotational cycle and each AEF is eligible for deployment for 120 days during that cycle. When one AEF’s deployment window ends and another’s begins, large amounts of personnel are deployed and redeployed as over 25,000 positions are vacated and filled. Planning these movements and tracking personnel throughout the process is a large part of the AEF Center’s responsibility.

(AF PD 10-4)

2.2 Air Force Deployments

The deployment planning process involves generating a Time Phased Force Deployment Document (TPFDD). A TPFDD lists which Unit Type Codes (UTCs) are required to be deployed in support of mission objectives for a given conflict, who will
supply these capabilities, and the timing and routing details of their transport. When a TPFDD is sourced, each UTC is assigned to a unit that will be tasked to fill the requirement. The TPFDD is used as a timeline for each step of the deployment of troops, cargo, and equipment. The information in the TPFDD is from the Joint Operations Planning and Execution System (JOPES) and is “pulled” into a TPFDD List (TPFDDL) when needed. The JOPES database includes all information pertaining to each requirement. A sample of the information contained in JOPES is shown below in Figure 1.

![Figure 1. Sample ULN Data from JOPES](image)

When pulling a TPFDD, desired parts of the data are chosen to form the different columns. Some sample lines from a TPFDDL are shown in Figure 2.

![Figure 2. Sample TPFDDL](image)
Figure 2. Sample TPFDDL

The TPFDDL above includes the OPLAN column which is the pseudo Plan ID (PID) for the operation planned by this TPFDDL, the REQUID or Requirement ID is also known as a Unit Line Number (ULN) which is a unique identifier for each tasking in a TPFDDL, the UTC code is listed next, and the Unit Information Code (UIC), which corresponds to the unit tasked to fill the UTC.

The travel timing columns are as follows: Ready to Load Date (RLD) when a unit is prepared to depart its origin, Available to Load Date (ALD) when a unit begins loading at the Point of Embarkation (POE), Earliest and Latest Arrival Dates (EAD and LAD) which are the earliest and latest dates a unit can arrive at its Point of Debarkation (POD)

<table>
<thead>
<tr>
<th>OPLAN</th>
<th>REQUID</th>
<th>TPFDDL DESCR</th>
<th>UIC</th>
<th>RLD</th>
<th>ALD</th>
<th>EAD</th>
<th>LAD</th>
<th>RDD</th>
<th>PAX</th>
<th>STONES</th>
<th>SERV</th>
<th>ORIG</th>
<th>POE</th>
<th>POD</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXXX</td>
<td>F031</td>
<td>3Z9A AVIATION</td>
<td>FFF</td>
<td>C02</td>
<td>C03</td>
<td>C05</td>
<td>C07</td>
<td>C09</td>
<td>1</td>
<td>01A HURBURY FL</td>
<td>BALTIMORE-WASH</td>
<td>B8BB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
and Required Delivery Date (RDD) which is the latest date a unit can arrive at its Forward Operating Location (FOL). The dates are in the format of the “C-Day” or “Conflict Day,” and C000 is the day the conflict begins and is the standard first day for a TPFDDL. Days beginning with N count backwards to N000, the day before the “C-Day.” The PAX and STONS columns list the number of passengers and short tons associated with each ULN. The SERV column indicates the MAJCOM of the tasked unit. The ORIG, POE, and POD columns give the origin, POE, and the POD for the ULN. (AFI 10-400) This research does not require any further information but there is much more available in JOPES; for example, destination, point of contact, line remarks, cargo information and airlift details (channel or dedicated) are all stored and available for those requiring such information as seen in figure 1. It is generally accepted in the planning community to call a TPFDDL a TPFDD and the terms are used interchangeably for the remainder of this research.

Large deployments made up of 100 or more passengers are usually given dedicated airlift by the TPFDD. Prior to January of 2005, smaller deployments were treated differently. The passengers were sent via commercial airlift to Baltimore Washington International (BWI) airport to board a “Patriot Express” contracted channel airlift mission for transportation to their POD where they would await further airlift to their FOL (AFI 10-403). Transportation planners at the PODs did not have visibility of the passengers until they were manifested on the Patriot Express flight and were on their way (~8-10 hours notice). This did not allow for proper scheduling of airlift from POD to FOL. Members would often have to wait up to 5 days at a POD before receiving tactical airlift to their FOL. (Hoog, 2005)
2.3 The Single Ticket Deployment Process

In January 2005, the Air Force began using a Single Ticket program for deploying personnel to the U.S. Central Command area of responsibility (the majority of our current deployed positions) (Hoog, 2005). In this process, all UTCs are given an itinerary for airlift from their home station to their final destination. To make this happen, In-Transit Visibility (ITV) and airlift forecasting were needed to ensure intra-theater airlift was staged properly to accommodate the large number of personnel traveling through PODs. This additional requirement was accomplished by a joint effort between Air Mobility Command (AMC) Headquarters, Tanker Airlift Control Center (TACC) and AEFC. (AEPE, 2005)

Single Tickets are particularly useful for personnel who require tactical airlift from their POD to their FOL. The goal of the Single Ticket process is to reduce the time a member spends at a POD waiting for airlift. With additional airlift forecasting and ITV, planners at the PODs receive notice of personnel transiting through their port well in advance and are able to schedule follow-on lift for shortly after their arrival, shortening the layover time awaiting forward deployment from as much as 5 days to as little as 24 hours. Decreasing this wait time speeds up the deployment of troops and shortens the time the airman is away from home station since travel time is not included in the 120 days of a deployment. Single Ticket deployments are a must in the eyes of the Combatant Commander as well as the service member (CENTAF, 2005).

By using the single ticket program for AEF 3/4, the Air Force successfully moved 3,173 deploying personnel through the PODs within 48 hours. Of these Airmen, 86% moved within 24 hours after arriving at their POD. This success came with much effort.
Planners responsible for the single ticket program had to manually track personnel and transmit ITV consuming an enormous amount of man-hours. Because the proper information was not in the usual online database, planners had to resort to faxes, and emails of spreadsheets or phone calls to notify the planners at the POD of incoming Airmen.

For AEF 5/6, 7,215 personnel deployed with a single ticket and 4473 of them required intratheater airlift from their POD to their FOL. Even though 3266 (73%) of these passengers received forward airlift within 24 hours, 1372 of them arrived at their POD on the wrong flight or without proper ITV. Without accurate ITV, POD planners have inaccurate or no information on the schedule or number of personnel arriving at their location and are unable to schedule adequate follow-on airlift (Hoog, 2005).

2.4 Aggregated Airlift

The Single Ticket deployment system is aided by Aggregated Airlift, grouping passengers together to justify dedicated airlift from POE to POD. To aggregate airlift, the TPFDD is sorted by POD and RDD to identify groups of passengers scheduled to go to the same POD in a relatively small time window (2-5 days). When a group of 100 or more is identified, then dedicated airlift is justified. Dedicated airlift can stop at more than one POE, provided there are at least 40 passengers at that location. Similarly, a single mission can deliver to more than one POD, provided there are at least 40 passengers going to each POD. No more than two POEs and no more than two PODs can be visited in a single mission. This aggregation facilitates reduced travel times by
allowing personnel to travel on dedicated airlift and gives planners at the POD the visibility and advance notice required to schedule tactical airlift to the FOL.

Aggregated airlift planning is currently completed manually using an Excel spreadsheet containing a sourced TPFDD. The process currently takes 5 working days and planners were able to aggregate 66% of the passengers for AEF 9/10. This research involves producing a more automated and efficient way to aggregate airlift by using an algorithm to search the TPFDD for feasible aggregates. By streamlining and improving the aggregation process, more passengers will be able to use dedicated airlift for the single ticket deployment process providing more efficient use of inter-theater airlift and leading to less travel time to and from deployed locations.

2.5 Airlift Aggregation as an Integer Program

The problem of airlift aggregation can be modeled as an integer program (IP) with the following formulation:

\[
\text{Min } Z = \sum_{j=1}^{n_1+n_2} N_i (1 - \sum_{j=1}^{m} X_{ij}) 
\]

(2.1)

Subject to the following constraints:

\[
X_{ij} \leq Y_j \quad j = 1 \text{ to } m \quad (2.2)
\]

\[
\sum_{j=1}^{m} X_{ij} = 1 \quad i = 1 \text{ to } n_1 + n_2 \quad (2.3)
\]

\[
\sum_{i=1}^{n_1+n_2} N_i X_{ij} \geq 100Y_j \quad j = 1 \text{ to } m \quad (2.4)
\]

\[
D \max_j \geq D_i X_{ij} \quad j = 1 \text{ to } m, \ i = 1 \text{ to } n_i \quad (2.5)
\]
\[ DRG_{\text{max}} \geq D_{i,j} \quad j = 1 \text{ to } m, \quad i = n_1 + 1 \text{ to } n_1 + n_2 \]  
\[ (2.6) \]

\[ D_{\text{min}} \leq D_{i,j} \quad j = 1 \text{ to } m, \quad i = 1 \text{ to } n_1 + n_2 \]  
\[ (2.7) \]

\[ DA_{\text{max}} - D_{\text{min}} \leq 5 \quad j = 1 \text{ to } m \]  
\[ (2.8) \]

\[ DRG_{\text{max}} - D_{\text{min}} \leq 2 \quad j = 1 \text{ to } m \]  
\[ (2.9) \]

\[ B_{i,j} = B_{i,k} \quad j = 1 \text{ to } m, \quad i = 1 \text{ to } n_1 + n_2, \quad k = 1 \text{ to } n_1 + n_2, \quad i \neq k \]  
\[ (2.10) \]

\[ P_{i,j} = P_{i,k} \quad j = 1 \text{ to } m, \quad i = 1 \text{ to } n_1 + n_2, \quad k = 1 \text{ to } n_1 + n_2, \quad i \neq k \]  
\[ (2.11) \]

Constants:

\[ m = \text{the total number of possible aggregates; } m \leq n_1 + n_2 \]

\[ n_1 = \text{the total number of Active Duty ULNs} \]

\[ n_2 = \text{the total number of Reserve and Guard ULNs} \]

\[ i = \text{the ULN index (The line the ULN is on in the TPFDD)} \]

\[ j = \text{the aggregate index (each aggregate requires a unique index)} \]

\[ N_i = \# \text{ of people in ULN } i \]

\[ D_i = \text{RDD for ULN } i \]

\[ B_i = \text{Region of origin for ULN } i \]

\[ P_i = \text{POD for ULN } i \]

Variables:

\[ X_{i,j} = 1 \text{ if ULN } i \text{ is in aggregate } j; \text{ 0 otherwise} \]

\[ Y_j = 1 \text{ if aggregate } j \text{ exists; } 0 \text{ otherwise} \]

\[ DRG_{\text{max}} = \text{the latest RDD for a Reserve or Guard ULN in aggregate } j \]
$D_{A \max_j} =$ the latest RDD for an Active Duty ULN in aggregate $j$

$D_{\min_j} =$ the earliest RDD in aggregate $j$

The objective function, Equation (2.1), sums all units not in an aggregate.

Equation (2.2) is the precedence constraint which forces $Y_j$ to 1 if aggregate $j$ is used.

Equation (2.3) ensures each unit will be assigned to only one aggregate. Equation (2.4) ensures each aggregate contains at least 100 personnel. Equations (2.5), (2.6), and (2.7) set the latest and earliest RDDs for Active and Reserve/Guard units in each aggregate (each aggregate will have only one earliest RDD). Equations (2.8) and (2.9) force the RDDs for active units to be within a 5 day span and reserve units to be within a 2 day span. The last two Equations (2.10) and (2.11) force aggregates to have a common region and point of debarkation.

The problem that arises in solving this formulation is that the number of ULNs in a TPFDD is usually at minimum 2,500. This IP has $4m + m(n_1+n_2)$ variables and $4m + 2mn_1 + 2mn_2 + n_1 + n_2 + 2m(n_1 + n_2 - 1)^2$ constraints. With a 7,500 passenger 2,500 line TPFDD containing 2,125 active duty ULNs and 375 active duty ULNs and a possible 75 aggregates (i.e. planning the maximum number of 100 passenger aggregates), the model would have 187,800 variables and over 937 million constraints. A commercial solver for a problem of this size would be a very costly alternative. In addition, many real world aspects of the problem are not included in this model, such as aggregates with multiple regions or PODs. Further combinations of aggregates could be made by the planners on a
case by case basis so the optimal solution of the model is not the final solution to the real problem but only a starting point.

2.6 The Hub Allocation Problem

This problem is considered generically by viewing the ULNs as any entity needing to be grouped for shipment. The home base is the origin, the POD is a hub, and the FOL is the destination. The RDD is the required delivery date of the entity. The Reserve/Guard or Active Duty attribute represents entities with different time windows with respect to how early the entities can arrive. In this form, the problem is similar to the p-median or hub allocation problem.

The p-median problem involves locating a set of hubs relative to a set of entities so as to minimize the sum of the distances or costs between all entities and their hub. The hub allocation problem involves the location of hubs in a network to minimize the number of hubs used and maximize coverage of the network (connect all entities to a hub). Given a network of consumers with varying constraints, the solution dictates how many hubs are used and where they are placed. The passengers or “consumers” are trying to get to a hub or POE compatible with their attributes. The McTRESH model proposed by Balakrishnan and Storbeck (1991), models maximum coverage with a threshold constraint. Current and Storbeck (1994) formulated a multiobjective model which selected hub locations based on the attributes of the network area in which they were to be placed. Constraints in their formulation force all hub locations to have a minimum number of customers. They also address the issue of the different demands of customers for a particular type of hub.
These formulations do not model individual consumers but an average or projection of their demands over a particular area and time span. The aggregated airlift problem is concerned with the exact demands of the ULNs in a particular TPFDD. A large part of the decision in a hub allocation problem is deciding where to locate a hub for continued operation and there is a cost associated with opening a hub and for maintaining its operation. The additional cost for the Air Force to “create” a hub is minimal as the bases and facilities already exist. Also the Air Force does not need to maintain the hub for future use. For example, if there are enough passengers to aggregate and leave from Dyess AFB, on a particular day going to the same POD(s), Dyess AFB becomes a hub. This may be the only time Dyess AFB is used as a hub, and the next cycle may use Lackland AFB. On the other hand if UPS wants to land a plane at an airport they do not normally use there, is an extra cost. UPS must consider the long term use of a hub based on the average demand for use of the facility.

2.7 Heuristic Search Methods

The airlift aggregation problem is known as a combinatorial optimization problem and as shown in the previous section, it is related to the p-median problem. The p-median problem is proven to belong to the class of NP-hard (nondeterministic polynomial-time complete) problems by Kariv and Halkimi (1979). It is generally accepted that there is no efficient algorithm which can solve these problems to optimality.

Heuristics are a way of finding high quality solutions to NP-hard problems in a short amount of time. Heuristics use “Rules of Thumb” and the context of the problem to construct and improve upon solutions to the problem. There are no guarantees of
optimality or solution quality with most heuristics; however, in many cases heuristics have performed well in practice.

A heuristic algorithm also allows us to model real world aspects of the problem with greater fidelity. For example, if there are more than 40 personnel departing from the same origin in the same time span, the AEFC planners first objective would be to aggregate them together. This is easily accomplished by a heuristic algorithm, but it would be difficult to model in an IP formulation.

Heuristics can be viewed as belonging to one of two categories: construction or local search. Constructive methods are generally greedy and deterministic in nature and build a single feasible solution. There are two approaches to this; primal which starts with an empty solution and builds the solution without violating primal feasibility; and dual, which starts with an infeasible solution and builds or destructs the solution until it is feasible. Local search heuristics start with a feasible (or infeasible) solution and move in some improving direction. A move is defined as an operation that changes one solution into another solution. A neighborhood of a solution is comprised of all solutions connected to the current solution via one move. Local search heuristics usually stop when they reach the local optima of their neighborhood (i.e. the current solution is better than all of its neighbors).

Metaheuristics are defined by Laguna as “a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality” (2000). They are usually a combination of a construction heuristics and a local search heuristics with the added ability to escape local optima, diversify their search into different neighborhoods, and intensify their search in desirable
neighborhoods. This research develops a multi-pass construction heuristic for the aggregated airlift problem.
3. Methodology

3.1 Overview

The algorithm used to construct a solution of feasible aggregates from a TPFDD is composed of three parts: generating a bound on the number of passengers able to be aggregated, creating an initial starting solution, and combining smaller aggregates to create aggregates with at least 100 passengers. The bound is created by identifying ULNs with RDDs which can never be assigned to an aggregate and subtracting them from the total amount of passengers. The initial starting solution places all ULNs in an aggregate but is infeasible because many aggregates do not contain the required 100 passengers. The starting solution is improved by combining infeasible aggregates with other aggregates, both feasible and infeasible, to create more feasible aggregates. The following sections provide model details.

3.2 Identifying Extraneous RDDs and Creating a Bound

The first step is to sort the TPFDD by RDD and remove the ULNs that will never be assigned to an aggregate. Each ULN can be a part of 5 aggregates as seen in Figure 3 below. The ULN can leave from 0 to 4 days early and therefore can be included with passengers in ULNs with RDDs up to 4 days earlier or later. RDD(r) is the day in question, the other days are those which can be included in the same aggregate as RDD(r). For example, if r = 30, the ULNs in RDD(r) are required on day 30. They can be aggregated with ULNs required on day 26 through 30 (r-4 to r), 27 through 31 (r-3 to r+1), 28 through 31 (r-2 to r+2), 29 through 32 (r-1 to r+3), or 30 through 34 (r to r+4).
Reserve/Guard passengers can only arrive up to 2 days early, so each aggregate can include Active Duty and Reserve/Guard passengers for the first 3 RDDs and only Active Duty for the last 2. Any ULNs that cannot be grouped with other ULNs totaling 100 passengers and leaving within 4 days before or after it (2 days for Reserve/Guard units) are removed from the list of candidates. For each RDD, the algorithm checks maximum number of passengers in all possible aggregates until it encounters an aggregate with more than 100 passengers. If a feasible aggregate can not be found for a particular RDD, all ULNs with that RDD are removed from consideration. When these ULNs are removed from consideration the model is left with an upper bound on the number of passengers it will be able to aggregate. The common region and POD constraints are not considered; therefore this bound is not expected it to be a tight bound.

### 3.3 Generating a Starting Solution

To generate a starting solution, each ULN is aggregated with others from the same origin going to the same POD within a feasible RDD span. Aggregates of ULNs with a common origin which are less than 75 are then disbanded so that they can be
aggregated with ULNs in the same region. A notional TPFDD, given below in Table 1, will be used to demonstrate the algorithm.

Table 1. Notional TPFDD

<table>
<thead>
<tr>
<th>Index</th>
<th>ULN</th>
<th>Origin</th>
<th># Pax</th>
<th>RDD</th>
<th>Reg</th>
<th>POD</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F9Q1X</td>
<td>ANDREWS</td>
<td>25</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>F933Q01</td>
<td>BEALE AF</td>
<td>51</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>F933R02</td>
<td>BEALE AF</td>
<td>49</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>F93BF01</td>
<td>DAVIS MO</td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>FT3A1</td>
<td>DAVIS MO</td>
<td>35</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>F933H01</td>
<td>EDWARDS</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>F933L01</td>
<td>EDWARDS</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>F03EX01</td>
<td>HILL AFB</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>F0HDX01</td>
<td>HILL AFB</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>F0HEX01</td>
<td>HILL AFB</td>
<td>43</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>F0HHX01</td>
<td>HILL AFB</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>FT638</td>
<td>LANGLEY</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>F9U1A</td>
<td>LANGLEY</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>F935B05</td>
<td>MOODY A</td>
<td>26</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>F99U3</td>
<td>MOODY A</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>F99U4</td>
<td>MOODY A</td>
<td>55</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>F96B6</td>
<td>MOODY A</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>F9PE5</td>
<td>MOODY A</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>F9702</td>
<td>NELLIS AF</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>F0PSW0</td>
<td>NELLIS AF</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>F9X3A</td>
<td>ROBINS A</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>FNJ1A01</td>
<td>ROBINS A</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>F9AHO1</td>
<td>SEYMOUR</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>F96EB01</td>
<td>SEYMOUR</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td>F99JK02</td>
<td>SEYMOUR</td>
<td>74</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>F99K02</td>
<td>SEYMOUR</td>
<td>55</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>27</td>
<td>F9HNB</td>
<td>SEYMOUR</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>28</td>
<td>F9HRH0</td>
<td>SEYMOUR</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>29</td>
<td>F0PSV0</td>
<td>WASHING</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>FNJ1A02</td>
<td>WASHING</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>31</td>
<td>F933G02</td>
<td>WRIGHT-F</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>F9NF001</td>
<td>WRIGHT-F</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>33</td>
<td>FT63601</td>
<td>WRIGHT-F</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>34</td>
<td>FT63602</td>
<td>WRIGHT-F</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>35</td>
<td>F96MD</td>
<td>BARKSDA</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>36</td>
<td>F99UA01</td>
<td>BEALE AF</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>37</td>
<td>F9N001</td>
<td>LANGLEY</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>38</td>
<td>F96F05</td>
<td>MOODY A</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>39</td>
<td>F9HB6</td>
<td>MOODY A</td>
<td>59</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>F03XM</td>
<td>RENO-TAH</td>
<td>13</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>41</td>
<td>F03XN</td>
<td>RENO-TAH</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>42</td>
<td>F0HD2</td>
<td>RENO-TAH</td>
<td>15</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>F98H0</td>
<td>SEYMOUR</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>F070602</td>
<td>TRAVIS AF</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
The algorithm starts with ULN(44) from Travis containing 1 Reserve/Guard passenger, designated as 2 in the service column, and compares it to the ULNs above it. There are no ULNs in the TPFDD with the same origin, so the algorithm stops searching when it reaches ULN(24) which is more than 2 days early and not in its feasible RDD span. Next the algorithm compares ULN(43) with those above it. The next earlier ULN with a common origin and POD is ULN(26), which is also within the feasible RDD span. ULN(43) and ULN(26) are updated with lastULN = 43 LastActiveRDD = 4, LastR/GRDD = 6 and totalpax = 65. Next ULN(25) is added and ULNs 43, 26, and, 25 are updated with totalpax = 139, and firstULN = 25. ULN(25) is updated with LastActiveRDD = 4, LastR/GRDD = 6, and lastULN = 44. The algorithm will stop when it reaches ULN(24) which is beyond the feasible RDD span for this aggregate since ULN(43) is Reserve/Guard.

After aggregating passengers from the same origin going to the same POD aggregates with less than 75 passengers are disbanded. The regional algorithm is then run to aggregate ULNs with a common POD and region. This phase would aggregate ULN(44) with ULNs 42, 41, 40, and 36 which all have a common region and POD and are within the feasible RDD span.

The regional algorithm stops adding to the aggregate when the number of passengers in the aggregate reaches 100 or more to keep the regional aggregates from leaving any earlier than necessary. Keeping large numbers of personnel from leaving too early is also the motivation for why the algorithm searches the TPFDD from the last ULN to the first ULN. For example, it would not be desirable for a 150 passenger ULN to
have to leave 3 days early to aggregate an additional 15 passengers. If possible, the additional passengers should leave with another aggregate required on an earlier day.

This portion of the algorithm completes the generation of an initial but likely infeasible solution. All ULNs are in an aggregate but many do not contain the required 100 passengers. A record is kept for those aggregates containing less than 100 passengers. The goal of the remaining passes of the heuristic is to group smaller aggregates together to create aggregates containing at least 100 passengers.

3.5 Combining Smaller Aggregates

Aggregates are combined based on the following priorities (provided by AEFC):

1. Aggregates containing passengers < 40 are combined with an aggregate in a region to the east containing passengers ≥ 100 going to the same POD.
2. Aggregates containing 40 ≤ passengers < 100 are combined with other 40 ≤ passengers < 100 aggregates from a different region but going to the same POD.
3. Aggregates containing 40 ≤ passengers < 100 are combined with other aggregates containing ≥ 100 passengers from a different region but going to the same POD.
4. Aggregates containing 40 ≤ passengers < 100 are combined with other aggregates containing 40 ≤ passengers < 100 from the same region but going to a different POD.
5. Aggregates containing $40 \leq$ passengers $< 100$ are combined with aggregates containing $\geq 100$ passengers from the same region but going to a different POD.

6. Aggregates containing $40 \leq$ passengers $< 100$ are combined with other aggregates containing $40 \leq$ passengers $< 100$ from a different region and going to a different POD.

7. Aggregates containing $40 \leq$ passengers $< 100$ are combined with aggregates containing $\geq 100$ passengers from a different region and going to a different POD.

Each phase can only combine aggregates which have not been combined in previous phases. This keeps aggregates from containing ULNs from more than 2 regions or more than 2 PODs. These phases are accomplished with the following pseudocode, broken down into phases. Each phase of the pseudocode differs only in the first two IF/THEN statements where the criteria of each phase are checked. If both aggregates meet the criteria, all ULNs belonging to the aggregates are combined into a single aggregate. Examples for each phase are presented on the pages that follow.
Figure 4. Phase 1 Combination

In figure 4 an aggregate with less than 40 passengers is combined with an aggregate with greater than 100 passengers to the east with the same POD. The regions are based on the time zones Pacific (1) to Eastern (4) so the greater region number is always to the east. The first aggregate contains 27 passengers from the west coast traveling to POD 4, it contains Active Duty passengers required on days 4 through 6 and Reserve/Guard passengers required on day 4. It is combined with the 104 passenger aggregate from the central region containing Active Duty passengers required on days 2 through 5 and Reserve/Guard passengers required on days 2 through 3. The combined aggregate will originate in the central region to be delivered through POD 4 by RDD 4 which will not be too early for the last Active Duty ULN required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.
In figure 5 two aggregates from different regions, each containing $40 \leq$ passengers $< 100$ and with a common POD are combined to form an aggregate with greater than 100 passengers. The first aggregate contains 44 passengers from the central region traveling to POD 4, it contains Active Duty passengers required on days 3 through 6 and Reserve/Guard passengers required on day 3. This aggregate is combined with the 57 passenger aggregate from the west coast region also traveling to POD 4. The 57 passenger aggregate contains Active Duty passengers required on days 2 through 5 and Reserve/Guard passengers required on days 2 through 4. The combined aggregate originates in the west coast region and stops in the central region. The combined aggregate is delivered through POD 4 by RDD 2 which is not too early for the last Active Duty ULN required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.
PHASE 3

In figure 6 an aggregate containing $40 \leq \text{passengers} < 100$ is combined with an aggregate with greater than 100 passengers going to the same POD but from a different region. The first aggregate contains 47 passengers from the central region traveling to POD 4 and includes Active Duty passengers required on days 3 through 6 and Reserve/Guard passengers required on day 3. It is combined with the 104 passenger aggregate from the west coast region containing Active Duty passengers also traveling to POD 4 who are required on days 2 through 5 and Reserve/Guard passengers required on days 2 through 4. The combined aggregate originates in the west coast region and stops in the central region. It is delivered through POD 4 by RDD 2 which is not too early for the last Active Duty ULN required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.
In figure 7 illustrates what happens when two aggregates containing $40 \leq$ passengers < 100 from a common region but going to different PODs are combined to form a cluster with greater than 100 passengers. The first aggregate contains 44 passengers from the west coast traveling to POD 4, and includes Active Duty passengers required on days 3 through 6 and Reserve/Guard passengers required on day 3. It is combined with the 57 passenger aggregate traveling to POD 2 from the west coast region containing Active Duty passengers required on days 2 through 5 and Reserve/Guard passengers required on days 2 through 4. The combined aggregate originates in the west coast region with delivery through POD 4 and POD 2 by RDD 2 which is not too early for the last
Active Duty ULN required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.

PHASE 5

47 Pax
ULN 8 to ULN 37
Region = 1
POD = 4
First RDD = 2
Last AD RDD = 5
Last R/G RDD = 4

+ 104 Pax
ULN 24 to ULN 57
Region = 1
POD = 2
First RDD = 3
Last AD RDD = 6
Last R/G RDD = 3

= 151 Pax
ULN 8 to ULN 57
Region = 1
Combined POD
First RDD = 2
Last AD RDD = 6
Last R/G RDD = 4

Figure 8. Phase 5 Combination

In figure 8 an aggregate with less than 40 passengers is combined with an aggregate with greater than 100 passengers from the same region but going to different PODs. The first aggregate contains 47 passengers from the west coast region traveling to POD 4, and includes Active Duty passengers required on days 2 through 5 and Reserve/Guard passengers required on day 4. It is combined with the 104 passenger aggregate also from the west coast region traveling to POD 2 containing Active Duty passengers required on days 3 through 6 and Reserve/Guard passengers required on day 3. The combined aggregate originates in the west coast region and is delivered through POD 4 and POD 2 by RDD 2.
which is not too early for the last Active Duty ULN required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.

**Figure 9. Phase 6 Combination**

In figure 9 two aggregates containing 40 ≤ passengers < 100 from different regions and going to different PODs are combined to form an aggregate with greater than 100 passengers. The first aggregate contains 44 passengers from the west coast region traveling to POD 4, and includes Active Duty passengers required on days 3 through 6 and Reserve/Guard passengers required on day 3. It is combined with the 57 passenger aggregate from the central region and contains Active Duty passengers traveling to POD 2 who are required on days 2 through 5 and Reserve/Guard passengers who are required on days 2 through 4. The combined aggregate originates in the west coast region, stops in the central region, and is delivered through POD 4 and POD 2 by RDD 2. RDD 2 is not too
early for the last Active Duty ULN required on day 6 or the last Reserve/Guard
ULN required on day 4. It contains ULNs between 8 and 57.

PHASE 7

47 Pax
ULN 24 to
ULN 57
Region=1
POD=2
First
RDD=3
Last AD
RDD=6
Last R/G
RDD=3

104 Pax
ULN 8 to
ULN 37
Region=3
POD=4
First
RDD=2
Last AD
RDD=5
Last R/G
RDD=4

151 Pax
ULN 8 to
ULN 57
Combined
Region
Combined
POD
First
RDD=2
Last AD
RDD=6
Last R/G
RDD=4

Figure 10. Phase 7 Combination

In figure 10 an aggregate containing $40 \leq \text{passengers} < 100$ is combined with an
aggregate with greater than 100 passengers from a different region and going to a
different POD. The first aggregate contains 47 passengers from the west coast
region traveling to POD 2, and includes Active Duty passengers required on days
3 through 6 and Reserve/Guard passengers required on day 3. It is combined with
the 104 passenger aggregate from the central region traveling to POD 4 that
contains Active Duty passengers required on days 2 through 5 and Reserve/Guard
passengers required on days 2 through 4. The combined aggregate originates in
the west coast region, stops in the central region, and is delivered through POD 4
and POD 2 by RDD 2 which is not too early for the last Active Duty ULN
required on day 6 or the last Reserve/Guard ULN required on day 4. It contains ULNs between 8 and 57.

3.6 Combining ULNs in Infeasible Aggregates

At this point, any aggregate containing less than 100 passengers is disbanded and the algorithm attempts to fit the unmatched ULNs into a feasible aggregate. For each unmatched ULN, the algorithm checks all feasible aggregates going to the ULN’s POD to find one with a feasible RDD span. At this point, the region of the aggregate and ULN is no longer considered since any remaining unmatched ULNs will fly out of the BWI Pax terminal in Baltimore on channel airlift.

3.7 Output and Post Processing

Post processing is accomplished by an analyst at AEFC who views the output and chooses a POE for each aggregate or a combination of aggregates. An example of the output is seen in Table 2. Each line of the output includes all of the aggregation information pertaining to a ULN. The ULN, origin, # passengers, and RDD are taken directly from the TPFDD. The region and POD are represented by a corresponding number. The service column shows whether the ULN is Active Duty (4) or Reserve/Guard (2). The ADRDD and RGRDD columns represent the latest RDD for Active Duty and Reserve/Guard ULNs, respectively, in the aggregate.

The Last and First ULN columns uniquely identify the aggregate by indicating the first and last ULN indices contained in the aggregate. The ULNs assigned to an aggregate are not necessarily listed continuously in the TPFDD; however, all ULNs with the same First and Last ULN values are assigned to the same aggregate allowing the
analyst to sort by these columns to display the aggregate. The total passenger count for
the aggregate is listed next, followed by the aggregated index 0 for greater than 100
passenger aggregates and 1 for less than 100 passenger aggregates. Finally, the First
RDD column indicates the earliest RDD of the aggregate. All passengers in the
aggregate must move as if their RDD were the same as the First RDD of the aggregate as
no ULNs can arrive late due to aggregation.
The algorithm constructs a solution of feasible aggregates from a TPFDD by generating a bound, creating an initial starting solution, and combining smaller aggregates to create aggregates with at least 100 passengers. The initial starting solution places all ULNs in an aggregate although many aggregates do not contain the required
100 passengers. These infeasible aggregates are combined with other aggregates, both feasible and infeasible, to create more feasible aggregates. The output includes all ULNs and their aggregation information.
4. Testing and Results

4.1 Introduction

This chapter presents the testing of the algorithm on a set of notional TPFDDs and two real world TPFDDs. The effect of each phase of the algorithm on aggregation is noted. The algorithm is also tested with and without regions enforced to see the effect this added constraint has on solution quality. The number of passengers in feasible aggregates over the number of passengers given by the upper bound is used to compare solution quality for all testing. However, many other factors, such as the division of the aggregates across regions, the maximum inclusion of passengers traveling from the same origin, and the number of aggregates generated could also be used to evaluate the solution quality.

4.2 Testing for performance on Notional TPFDDs

Due to the complexity involved in declassifying a TPFDD, this research only had access to three actual TPFDDs, one of which was on a classified system and could not be fully analyzed. Therefore, a process to randomly generate notional TPFDDs was generated using the real world TPFDDs as seeds.

Since the TPFDD used is for rotation of personnel into required positions, it is assumed the actual requirements of these positions do not change significantly. It is also assumed there is some variability in the RDDs for each position from rotation to rotation and large amounts of variability in the origin as different units are tapped to fill the
requirements. The RDDs for each tasking were randomly increased or decreased by up to 3 days. The origins were selected from 80 origins with 20 in each region to evenly space taskings across the four regions. Analysis of the two fully sourced TPFDDs show 14% and 16% of taskings are filled by Reserve/Guard personnel; therefore, a ratio of 15% Reserve/Guard to 85% Active duty taskings was used to generate the notional TPFDDs. To accomplish this, 12 of the 80 origins were designated as Guard/Reserve origins. An origin was then randomly selected, based on a uniform distribution, to fill each ULN. The two real world TPFDDS were from 12 October (AEF 7/8) and 12 January (AEF 9/10). Notional TPFDDs 1-15 were generated from the 12 January TPFDD and 16-30 were generated from the 12 October TPFDD.

The first test returns the contribution of each phase of the algorithm to determine if all stages aggregate at least some ULNs. A percentage was gathered after the initial starting solution was generated (initial), after < 40 passenger aggregates were grouped with feasible aggregates (Phase 1), after > 40 passenger aggregates were grouped with each other and then feasible aggregates from different regions (Phases 2 and 3), with different PODs (Phases 4 and 5), then different regions and PODs (Phases 6 and 7), and after attempting to group all unmatched ULNs with feasible aggregates (Final Phase). Results of this testing can be seen in Table 3.
<table>
<thead>
<tr>
<th>TPFDD #</th>
<th>Initial</th>
<th>1</th>
<th>2/3</th>
<th>4/5</th>
<th>6/7</th>
<th>Final</th>
<th>Totalpax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23.71</td>
<td>30.45</td>
<td>70.61</td>
<td>72.57</td>
<td>74.79</td>
<td>86.28</td>
<td>97.28</td>
</tr>
<tr>
<td>2</td>
<td>20.36</td>
<td>27.14</td>
<td>70.06</td>
<td>72.51</td>
<td>77.52</td>
<td>85.69</td>
<td>96.71</td>
</tr>
<tr>
<td>3</td>
<td>19.09</td>
<td>23.75</td>
<td>68.07</td>
<td>72.25</td>
<td>72.80</td>
<td>86.23</td>
<td>97.08</td>
</tr>
<tr>
<td>4</td>
<td>21.69</td>
<td>28.36</td>
<td>69.02</td>
<td>71.77</td>
<td>72.93</td>
<td>90.23</td>
<td>98.22</td>
</tr>
<tr>
<td>5</td>
<td>21.37</td>
<td>24.69</td>
<td>69.78</td>
<td>72.98</td>
<td>73.61</td>
<td>88.68</td>
<td>98.75</td>
</tr>
<tr>
<td>6</td>
<td>23.88</td>
<td>27.21</td>
<td>74.00</td>
<td>75.53</td>
<td>76.79</td>
<td>87.11</td>
<td>98.33</td>
</tr>
<tr>
<td>7</td>
<td>23.75</td>
<td>28.34</td>
<td>66.82</td>
<td>69.52</td>
<td>71.65</td>
<td>84.52</td>
<td>97.41</td>
</tr>
<tr>
<td>8</td>
<td>16.10</td>
<td>27.19</td>
<td>66.76</td>
<td>71.86</td>
<td>73.13</td>
<td>87.41</td>
<td>98.51</td>
</tr>
<tr>
<td>9</td>
<td>24.97</td>
<td>29.49</td>
<td>69.89</td>
<td>73.47</td>
<td>73.47</td>
<td>89.39</td>
<td>96.85</td>
</tr>
<tr>
<td>10</td>
<td>24.51</td>
<td>32.47</td>
<td>72.74</td>
<td>76.84</td>
<td>77.93</td>
<td>89.84</td>
<td>97.10</td>
</tr>
<tr>
<td>11</td>
<td>20.34</td>
<td>28.15</td>
<td>70.51</td>
<td>72.80</td>
<td>75.16</td>
<td>89.88</td>
<td>97.92</td>
</tr>
<tr>
<td>12</td>
<td>24.55</td>
<td>27.93</td>
<td>68.83</td>
<td>71.57</td>
<td>73.33</td>
<td>88.27</td>
<td>97.09</td>
</tr>
<tr>
<td>13</td>
<td>17.23</td>
<td>26.32</td>
<td>71.97</td>
<td>74.99</td>
<td>76.84</td>
<td>92.45</td>
<td>96.15</td>
</tr>
<tr>
<td>14</td>
<td>23.30</td>
<td>28.89</td>
<td>68.40</td>
<td>72.21</td>
<td>72.68</td>
<td>90.09</td>
<td>98.28</td>
</tr>
<tr>
<td>15</td>
<td>29.70</td>
<td>31.96</td>
<td>66.89</td>
<td>75.43</td>
<td>77.81</td>
<td>91.94</td>
<td>97.53</td>
</tr>
<tr>
<td>16</td>
<td>38.65</td>
<td>40.03</td>
<td>66.95</td>
<td>70.78</td>
<td>70.78</td>
<td>82.03</td>
<td>97.83</td>
</tr>
<tr>
<td>17</td>
<td>30.95</td>
<td>36.54</td>
<td>67.82</td>
<td>68.98</td>
<td>68.98</td>
<td>77.81</td>
<td>96.90</td>
</tr>
<tr>
<td>18</td>
<td>29.77</td>
<td>39.51</td>
<td>65.42</td>
<td>67.28</td>
<td>67.28</td>
<td>76.31</td>
<td>97.44</td>
</tr>
<tr>
<td>19</td>
<td>33.66</td>
<td>36.47</td>
<td>66.75</td>
<td>68.07</td>
<td>68.87</td>
<td>78.50</td>
<td>96.57</td>
</tr>
<tr>
<td>20</td>
<td>31.47</td>
<td>37.13</td>
<td>69.06</td>
<td>69.06</td>
<td>69.66</td>
<td>80.46</td>
<td>97.52</td>
</tr>
<tr>
<td>21</td>
<td>35.51</td>
<td>38.30</td>
<td>64.60</td>
<td>66.96</td>
<td>66.96</td>
<td>79.71</td>
<td>97.19</td>
</tr>
<tr>
<td>22</td>
<td>36.31</td>
<td>41.90</td>
<td>69.27</td>
<td>70.69</td>
<td>71.26</td>
<td>80.43</td>
<td>96.92</td>
</tr>
<tr>
<td>23</td>
<td>27.69</td>
<td>36.10</td>
<td>64.48</td>
<td>67.03</td>
<td>67.67</td>
<td>82.27</td>
<td>96.77</td>
</tr>
<tr>
<td>24</td>
<td>32.79</td>
<td>38.34</td>
<td>62.63</td>
<td>67.31</td>
<td>67.31</td>
<td>81.50</td>
<td>97.72</td>
</tr>
<tr>
<td>25</td>
<td>34.70</td>
<td>41.67</td>
<td>65.20</td>
<td>68.61</td>
<td>68.61</td>
<td>77.70</td>
<td>97.30</td>
</tr>
<tr>
<td>26</td>
<td>34.20</td>
<td>39.73</td>
<td>63.62</td>
<td>65.15</td>
<td>65.78</td>
<td>79.44</td>
<td>97.95</td>
</tr>
<tr>
<td>27</td>
<td>33.08</td>
<td>41.47</td>
<td>66.69</td>
<td>68.92</td>
<td>68.92</td>
<td>81.71</td>
<td>96.96</td>
</tr>
<tr>
<td>28</td>
<td>31.46</td>
<td>37.04</td>
<td>67.61</td>
<td>70.09</td>
<td>70.82</td>
<td>78.68</td>
<td>97.17</td>
</tr>
<tr>
<td>29</td>
<td>29.64</td>
<td>39.38</td>
<td>65.02</td>
<td>68.81</td>
<td>68.81</td>
<td>78.92</td>
<td>97.42</td>
</tr>
<tr>
<td>30</td>
<td>28.43</td>
<td>36.76</td>
<td>64.32</td>
<td>68.27</td>
<td>68.27</td>
<td>76.00</td>
<td>97.61</td>
</tr>
<tr>
<td>Average</td>
<td>27.43</td>
<td>33.42</td>
<td>67.79</td>
<td>70.74</td>
<td>71.51</td>
<td>83.98</td>
<td>97.42</td>
</tr>
<tr>
<td>Max</td>
<td>38.65</td>
<td>41.90</td>
<td>74.00</td>
<td>76.84</td>
<td>77.93</td>
<td>92.45</td>
<td>98.75</td>
</tr>
<tr>
<td>Min</td>
<td>16.10</td>
<td>23.75</td>
<td>62.63</td>
<td>65.15</td>
<td>65.78</td>
<td>76.00</td>
<td>96.15</td>
</tr>
</tbody>
</table>

Each phase of aggregation did contribute to the final solution. Eliminating a phase could possibly lead to the same level of aggregation or even higher since some can
be viewed as a restriction. However, the phases are used to enforce the priorities of the decision maker and therefore provide functionality beyond the overall aggregation rate.

The algorithm performed better on the TPFDDs created using the 12 January TPFDD than those created from the 12 October TPFDD. These TPFDDs differ in the tightness of the movement window which is discussed further in Section 4.4. Overall, the algorithm performed well and significantly outperformed the manual aggregation percentage of 66%. A 95% confidence interval was calculated for the results using the interval estimator for quantiles presented by Banks (2005) using Equation (4.1) below:

\[
\hat{p} \pm z_{a/2} \sqrt{\frac{\hat{p}(1 - \hat{p})}{R - 1}}
\]  

(4.1)

This equation takes into account that the results are a proportion. The confidence intervals for performance are presented in Table 4.

<table>
<thead>
<tr>
<th>Confidence Interval on the final Aggregation %</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval Upper</td>
<td>97.33</td>
</tr>
<tr>
<td>Interval Lower</td>
<td>70.63</td>
</tr>
</tbody>
</table>

### 4.3 Testing for Performance on Real World TPFDDs

The algorithm was also tested on two real world TPFDDs. As seen below in Table 5, the algorithm was able to aggregate a greater percentage of the passengers than in the notional TPFDDs. The different performance is due to the fact that the notional TPFDDs were randomly sourced. Sourcing performed at AEFC is not done randomly. As seen in the small sample TPFDD in Chapter 2, some UTCs originate in the same location with
the same POD and RDD. This makes a logically sourced TPFDD more likely to have a higher aggregation percentage than a randomly sourced TPFDD. The 12 October TPFDD is from AEF cycle 7/8 of which 64% of the passengers were manually aggregated. The 12 January TPFDD is from AEF cycle 9/10 of which 66% of the passengers were manually aggregated.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial</th>
<th>1</th>
<th>2/3</th>
<th>4/5</th>
<th>6/7</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Pax</td>
<td>2757</td>
<td>4687</td>
<td>6037</td>
<td>6079</td>
<td>6194</td>
<td>6504</td>
</tr>
<tr>
<td>% Bound</td>
<td>38.12</td>
<td>64.81</td>
<td>83.48</td>
<td>84.06</td>
<td>85.65</td>
<td>89.93</td>
</tr>
<tr>
<td>% Total</td>
<td>37.37</td>
<td>63.54</td>
<td>81.84</td>
<td>82.40</td>
<td>83.96</td>
<td>88.17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial</th>
<th>1</th>
<th>2/3</th>
<th>4/5</th>
<th>6/7</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Pax</td>
<td>6789</td>
<td>6870</td>
<td>8544</td>
<td>8608</td>
<td>8653</td>
<td>8772</td>
</tr>
<tr>
<td>% Bound</td>
<td>74.71</td>
<td>75.60</td>
<td>94.02</td>
<td>94.73</td>
<td>95.22</td>
<td>96.53</td>
</tr>
<tr>
<td>% Total</td>
<td>74.20</td>
<td>75.08</td>
<td>93.38</td>
<td>94.08</td>
<td>94.57</td>
<td>95.87</td>
</tr>
</tbody>
</table>

The 12 October TPFDD has 1773 less passengers than the 12 January TPFDD. Both TPFDDs have the same number of in-cycle AEF rotation personnel. The difference is in the amount of out of cycle personnel or those personnel on a 6-month rotation versus the normal 4-month AEF rotation. The 12 October TPFDD has less out of cycle personnel and has RDDs spread out farther than those in the 12 January TPFDD. This increase in the number of personnel traveling in the first 30 days of the TPFDD allowed for more aggregation in the 12 January TPFDD. Figure 11 illustrates the difference in the two TPFDDs. The 12 October TPFDD has taskings spread out over a wider range of RDDs than the 12 January TPFDD. There are no ULNs after RDD 33 that could be assigned to an aggregate in the 12 January TPFDD. However, after RDD 33, there are only 25 ULNs containing a total of 63 passengers – a small loss in the final percentage.
Conversely, there are 90 ULNs containing a total 274 passengers after RDD 33 in the 12 October TPFDD with some ULNs as late as RDD 55. TPFDDS with all passengers in a relatively short span of RDDs allows more of them to be grouped together into feasible aggregates.

![Passenger Distributions of TPFDDs](image)

**Figure 11. Passenger Distributions of Real World TPFDDs**

4.4 Analysis of Removing the Regions Constraint

Finally, the effect of the region constraints on the solution is tested. To accomplish this, all ULNs were labeled as being in one region. Of course, phases 2/3 and 6/7 were had no effect as they are based on the combination of regions. The increase in percentage of aggregation is seen in Tables 6 and 7. As expected, removing the region constraint increased the aggregation percentage, reaching the optimal value of 100% of the bound on one TPFDD. When the region constraints are removed, the goal of aggregation and ITV is still achieved; however, the personnel may be required to utilize significantly more commercial airlift to get to their POE.
Table 6. Aggregation Results of Notional TPFDDs without Region Constraints

<table>
<thead>
<tr>
<th>TPFDD</th>
<th>Initial</th>
<th>1</th>
<th>4/5</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46.512</td>
<td>70.104</td>
<td>89.147</td>
<td>99.079</td>
</tr>
<tr>
<td>2</td>
<td>53.927</td>
<td>72.008</td>
<td>91.796</td>
<td>98.576</td>
</tr>
<tr>
<td>3</td>
<td>54.723</td>
<td>69.357</td>
<td>94.743</td>
<td>100.000</td>
</tr>
<tr>
<td>4</td>
<td>51.641</td>
<td>68.332</td>
<td>90.531</td>
<td>97.307</td>
</tr>
<tr>
<td>5</td>
<td>52.933</td>
<td>71.746</td>
<td>93.515</td>
<td>97.665</td>
</tr>
<tr>
<td>6</td>
<td>43.992</td>
<td>69.557</td>
<td>91.619</td>
<td>98.333</td>
</tr>
<tr>
<td>7</td>
<td>54.796</td>
<td>72.748</td>
<td>92.606</td>
<td>98.463</td>
</tr>
<tr>
<td>8</td>
<td>50.566</td>
<td>71.644</td>
<td>95.329</td>
<td>99.079</td>
</tr>
<tr>
<td>9</td>
<td>54.694</td>
<td>72.749</td>
<td>91.864</td>
<td>99.413</td>
</tr>
<tr>
<td>10</td>
<td>52.977</td>
<td>73.236</td>
<td>93.438</td>
<td>98.762</td>
</tr>
<tr>
<td>11</td>
<td>45.759</td>
<td>68.080</td>
<td>91.484</td>
<td>97.288</td>
</tr>
<tr>
<td>12</td>
<td>47.479</td>
<td>73.368</td>
<td>94.169</td>
<td>99.381</td>
</tr>
<tr>
<td>13</td>
<td>52.194</td>
<td>68.106</td>
<td>94.442</td>
<td>98.943</td>
</tr>
<tr>
<td>14</td>
<td>49.416</td>
<td>70.544</td>
<td>91.771</td>
<td>98.021</td>
</tr>
<tr>
<td>15</td>
<td>55.087</td>
<td>71.896</td>
<td>92.346</td>
<td>98.375</td>
</tr>
<tr>
<td>16</td>
<td>44.520</td>
<td>61.147</td>
<td>81.779</td>
<td>91.811</td>
</tr>
<tr>
<td>17</td>
<td>50.518</td>
<td>63.109</td>
<td>79.337</td>
<td>94.306</td>
</tr>
<tr>
<td>18</td>
<td>51.906</td>
<td>61.644</td>
<td>77.476</td>
<td>93.934</td>
</tr>
<tr>
<td>19</td>
<td>42.673</td>
<td>63.728</td>
<td>83.156</td>
<td>92.701</td>
</tr>
<tr>
<td>20</td>
<td>55.796</td>
<td>64.137</td>
<td>83.611</td>
<td>93.022</td>
</tr>
<tr>
<td>21</td>
<td>51.325</td>
<td>63.877</td>
<td>82.510</td>
<td>91.660</td>
</tr>
<tr>
<td>22</td>
<td>45.776</td>
<td>65.357</td>
<td>83.972</td>
<td>97.455</td>
</tr>
<tr>
<td>23</td>
<td>48.186</td>
<td>64.995</td>
<td>83.023</td>
<td>94.509</td>
</tr>
<tr>
<td>24</td>
<td>48.218</td>
<td>62.089</td>
<td>78.166</td>
<td>93.439</td>
</tr>
<tr>
<td>25</td>
<td>43.982</td>
<td>67.665</td>
<td>81.583</td>
<td>94.372</td>
</tr>
<tr>
<td>26</td>
<td>52.491</td>
<td>63.562</td>
<td>81.843</td>
<td>92.306</td>
</tr>
<tr>
<td>27</td>
<td>54.383</td>
<td>69.761</td>
<td>84.887</td>
<td>93.499</td>
</tr>
<tr>
<td>28</td>
<td>45.801</td>
<td>68.122</td>
<td>84.208</td>
<td>91.057</td>
</tr>
<tr>
<td>29</td>
<td>53.889</td>
<td>66.412</td>
<td>85.056</td>
<td>93.238</td>
</tr>
<tr>
<td>30</td>
<td>49.924</td>
<td>61.033</td>
<td>80.017</td>
<td>93.418</td>
</tr>
<tr>
<td>Average</td>
<td>50.203</td>
<td>67.670</td>
<td>87.314</td>
<td>95.980</td>
</tr>
<tr>
<td>Max</td>
<td>55.796</td>
<td>73.368</td>
<td>95.329</td>
<td>100.000</td>
</tr>
<tr>
<td>Min</td>
<td>42.673</td>
<td>61.033</td>
<td>77.476</td>
<td>91.057</td>
</tr>
</tbody>
</table>
The real world TPFDDs were also aggregated without the region constraints and the results are seen in Table 7. The aggregation percentages increased by 2-5% for the real world TPFDDs. However, without regions to guide the formation of aggregates, they were widespread across the CONUS and would be difficult to use. The difference can be seen in a sample output from the aggregation of the 12 October TPFDD with and without regions in Table 8.
### Table 8. Sample Output With and Without Region Constraint

<table>
<thead>
<tr>
<th>With Region Constraint</th>
<th>Without Region Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Origin</strong></td>
<td><strong># Pax</strong></td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>4</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>4</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>7</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>11</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>1</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>4</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>2</td>
</tr>
<tr>
<td>EDWARDS AFB</td>
<td>6</td>
</tr>
<tr>
<td>EDWARDS AFB</td>
<td>17</td>
</tr>
<tr>
<td>EDWARDS AFB</td>
<td>16</td>
</tr>
<tr>
<td>HILL AFB</td>
<td>10</td>
</tr>
<tr>
<td>HILL AFB</td>
<td>5</td>
</tr>
<tr>
<td>HILL AFB</td>
<td>43</td>
</tr>
<tr>
<td>HILL AFB</td>
<td>1</td>
</tr>
<tr>
<td>DAVIS MONTAN AFB</td>
<td>1</td>
</tr>
<tr>
<td>NELLIS AFB</td>
<td>1</td>
</tr>
<tr>
<td>TRAVIS AFB</td>
<td>1</td>
</tr>
<tr>
<td>NELLIS AFB</td>
<td>2</td>
</tr>
<tr>
<td>TRAVIS AFB</td>
<td>2</td>
</tr>
<tr>
<td>BEALE AFB</td>
<td>1</td>
</tr>
<tr>
<td>NELLIS AFB</td>
<td>1</td>
</tr>
<tr>
<td>NELLIS AFB</td>
<td>1</td>
</tr>
<tr>
<td>NELLIS AFB</td>
<td>1</td>
</tr>
</tbody>
</table>

The aggregate generated using the region constraint includes units in the Pacific and Mountain time zones. The aggregate generated without the region constraint is mostly composed of units on the west coast but also includes personnel from all other regions. The 2 passengers from Andrews AFB, for example, would have to fly to the west coast to join the aggregate. This, of course, is undesirable as AEFC would like to minimize the number of passengers having to fly east to west. The other option would be for the aggregate to have a POE on the east coast which would cause 137 passengers to fly commercial or separately contracted flights to the POE. Based on the limited gain in
aggregation in the real world TPFDDs and the preferences of the planners at AEFC, it was decided to use the region constraints.

The testing of the algorithm was performed on a set of notional TPFDDS and two real world TPFDDs. The effect of each phase of the algorithm on aggregation was noted. The effect of the region constraint on solution quality was tested as well. Conclusions drawn about these results, the impact of this research, its current application, and further research opportunities are discussed in Chapter 5.
5. Conclusion

5.1 Summary of Results

The algorithm performed well on the randomly sourced notional TPFDDs and better on the real world TPFDDs outperforming the manual process by more than 25%. It aggregates a 2500 line TPFDD in approximately 1 minute on a 1.5 GHz PC. In most instances, each phase of the algorithm is utilized to enforce the priorities of the decision maker providing a high aggregation percentage while meeting their desires in terms of the content of the aggregates. Forcing aggregates to conform to regions degrades the aggregation percentage slightly but results in aggregates that can be employed more efficiently and for less cost.

5.2 Airlift Aggregation for AEF Cycle 1/2

In January 2006, the algorithm was used to aggregate passengers for AEF cycle 1/2. Of the 5425 sourced personnel positions in the TPFDD, the algorithm was able to aggregate 4673 or 86.14%. After the remainder of the TPFDD is sourced, we believe the percentage of aggregation will increase. The AEFC planner was able to use these results to dramatically improve the airlift aggregation process for AEF Cycle 1/2 and plans to use the algorithm on future AEF Cycles as well.

A 20% increase in aggregation percentage translates to 1,850 additional airmen flying dedicated airlift from their home station (or a POE in their region) to their POD each rotation or 3 times per year. The additional ITV for these airmen means they will receive intratheater airlift within 24 hours of arrival to the POD instead of the usual 72-
hour layover. The result is more airmen are provided to the Combatant Commander in a timelier manner and at less cost and with less time spent away from home.

5.3 Further Research

Further research in the area of airlift aggregation can be broken down into two categories, algorithm improvement and analysis of the constraints. The algorithm could be improved upon to select a POE for each aggregate from a list of designated military installations satisfying the necessities of a POE for aggregated airlift such as sufficient lodging, proximity to a major airport, and adequate passenger handling capability. The selection could be made based on minimizing the cost of transporting all of the aggregate’s ULNs to the POE.

Parameters such as feasible RDD spans, minimum number of passengers required for dedicated airlift and minimum number of passengers required for a POE/POD were all provided by the decision maker. The values of these parameters could dramatically affect algorithm performance. Through decision and sensitivity analysis we could other alternative values and the impact on the aggregation rate.
Bibliography


Appendix A: VBA Code

The following is the VBA code run with Microsoft Excel:

Sub RDDbound()

L = 2466

N = 105

' This sequence saves the number of active duty troops leaving on RDD i in RDDset(i,0),
' the number of reserve guard troops in RDDset(i,1),
' and the total number of troops in RDDset(i,2)

For o = 0 To 14

Dim RDDset() As Integer, Totalcluster As Integer, RDDAvg() As Integer
Dim Keep() As Boolean, reserveguard() As Boolean

ReDim RDDset(N, 3), Keep(N), reserveguard(N), RDDAvg(N)

Sheets("CONUS B").Select

For i = 0 To N

Selection.AutoFilter Field:=5 * o + 1, Criteria1:=i

RDDset(i, 2) = Cells(L + 1, 5 * o + 2).Value

Selection.AutoFilter Field:=5 * o + 4, Criteria1:=4

RDDset(i, 0) = Cells(L + 1, 5 * o + 2).Value

Selection.AutoFilter Field:=5 * o + 4

Selection.AutoFilter Field:=5 * o + 4, Criteria1:=2

RDDset(i, 1) = Cells(L + 1, 5 * o + 2).Value

Selection.AutoFilter Field:=5 * o + 4

Next i

' Check RDD sets in the tails to see if they can be clustered

For y = 0 To 3

Selection.AutoFilter Field:=5 * o + 4
Selection.AutoFilter Field:=5 * o + 1

Next y

End Sub
If RDDset(0 + y, 0) + RDDset(1 + y, 0) + RDDset(2 + y, 0) + RDDset(3 + y, 0) + RDDset(4 + y, 0) + RDDset(0 + y, 1) + RDDset(1 + y, 1) + RDDset(2 + y, 1) + RDDset(3 + y, 1) > 100 Then
   Keep(0 + y) = True
   Keep(1 + y) = True
   Keep(2 + y) = True
   Keep(3 + y) = True
   Keep(4 + y) = True
End If
Next y

For z = 0 To 3
   If RDDset(N - 4 - z, 0) + RDDset(N - 3 - z, 0) + RDDset(N - 2 - z, 0) + RDDset(N - 1 - z, 0) + RDDset(N - z, 0) + RDDset(N - 4 - z, 1) + RDDset(N - 3 - z, 1) + RDDset(N - 2 - z, 1) + RDDset(N - 1 - z, 1) + RDDset(N - z, 1) > 100 Then
      Keep(N - 4 - z) = True
      Keep(N - 3 - z) = True
      Keep(N - 2 - z) = True
      Keep(N - 1 - z) = True
      Keep(N - z) = True
   End If
Next z

'Check RDD sets in the middle to see if they can be clustered

For S = 4 To N - 4
   If RDDset(S, 0) > 100 Then
      Keep(S) = True
   Else
      If RDDset(S, 2) = 0 Then
         Keep(S) = False
      Else
         For d = 0 To 4
            If RDDset((S - (4 - d)), 0) + RDDset((S - (3 - d)), 0) + RDDset((S - (2 - d)), 0) + RDDset((S - (1 - d)), 0) + RDDset((S + d), 0) + RDDset((S - (4 - d)), 1) + RDDset((S - (3 - d)), 1) + RDDset((S - (2 - d)), 1) + RDDset((S - (1 - d)), 1) > 100 Then
               Keep(S) = True
            End If
         Next d
      End If
   End Else
Next S
For x = 1 To N
    RDDAvg(x) = RDDAvg(x) + RDDset(x, 2)
Next x

'Output Results of RDD sets that can possibly be clustered, this is used as a bound
Totalcluster = 0
totalnotcluster = 0
x = 0
For b = 0 To N
    If Keep(b) = True Then
        Worksheets("Data").Range("A1").Offset(x + 1, 5 * o) = b
        Worksheets("Data").Range("A1").Offset(x + 1, 5 * o + 1) = RDDset(b, 2)
        Worksheets("Data").Range("A1").Offset(x + 1, 5 * o + 2) = RDDset(b, 0)
        Worksheets("Data").Range("A1").Offset(x + 1, 5 * o + 3) = RDDset(b, 1)
        x = x + 1
        Totalcluster = Totalcluster + RDDset(b, 2)
    Else
        totalnotcluster = totalnotcluster + RDDset(b, 2)
    End If
Next b
Worksheets("Data").Range("A1").Offset(0, 5 * o + 4) = "Total Pax Removed"
Worksheets("Data").Range("A1").Offset(1, 5 * o + 4) = totalnotcluster
Worksheets("Data").Range("A1").Offset(5, 5 * o + 4) = "Total Pax to Cluster"
Worksheets("Data").Range("A1").Offset(6, 5 * o + 4) = Totalcluster
Next o
For b = 0 To N
    Worksheets("Data2").Range("A1").Offset(b, 0) = RDDAvg(N) / 15
Next b
End Sub
Sub Aggregation()
'
' Macro3 Macro
' Macro recorded 12/6/2005 by AFIT
'
N = 105        'the number of RDD's in the TPFDD
L = 2466       'the number of lines in the TPFDD

Dim ULNL() As Integer, ULNI() As Integer
Dim FirstULN, LastULN As Integer
Dim v() As Integer

o = 1 '(this can be used as a loop for multiple runs)

ReDim ULNI(L, 12), ULNL(L, 2), v(8)

' This sequence will build a ULN matrix with ULN, #Pax, RDD, Region, POD, Active or
' Reserve/Guard,
' and Earliest RDD in its cluster
' ULNL includes line number and Origin
' ULNI includes #Pax, RDD, Region, Active Duty or Guard, and the Cluster that the ULN
' is in

Sheets("CONUS B").Select
Range("A1").Select

For j = 2 To L - 2
    If Cells(j, 5 * o + 2) > 0 Then
        region = Cells(j, 5 * o + 3).Value / 20
        ULNI(j, 0) = Cells(j, 5 * o + 2).Value
        ULNI(j, 1) = Cells(j, 5 * o + 1).Value
        ULNL(j, 1) = Cells(j, 5 * o + 3).Value
        ULNL(j, 2) = Cells(j, 5 * o + 3).Value
        ULNI(j, 2) = region
        ULNI(j, 3) = Cells(j, 5 * o + 5).Value
        ULNI(j, 4) = Cells(j, 5 * o + 4).Value
    End If

Next j
This sequence clusters ULNs with the same origin and POD leaving in a feasible RDD span.
Clusters < 40 are kept and others are broken down to ULNs.

For \( g = 1 \) To 5

For \( k = 0 \) To \( L \)
If \( \text{ULNI}(L - k, 3) = g \) And \( \text{ULNI}(L - k, 0) \neq 0 \) And \( \text{ULNI}(L - k, 9) = 0 \) Then
  \( \text{ULNI}(L - k, 7) = L - k \)
  \( \text{ULNI}(L - k, 8) = L - k \)
  \( \text{ULNI}(L - k, 9) = \text{ULNI}(L - k, 0) \)
  If \( \text{ULNI}(L - k, 4) = 2 \) Then
    \( \text{ULNI}(L - k, 5) = \text{ULNI}(L - k, 1) \)
  End If
  If \( \text{ULNI}(L - k, 4) = 4 \) Then
    \( \text{ULNI}(L - k, 6) = \text{ULNI}(L - k, 1) \)
  End If
End If

For \( r = k + 1 \) To \( L \)
If \( r \neq k \) And \( \text{ULNL}(L - k, 2) = \text{ULNL}(L - r, 2) \) And \( \text{ULNI}(L - r, 3) = g \) And \( \text{ULNI}(L - r, 7) = 0 \) Then
  If \( \text{ULNI}(L - k, 6) - \text{ULNI}(L - r, 1) \leq 4 \) And \( \text{ULNI}(L - k, 5) - \text{ULNI}(L - r, 1) \leq 2 \) Then
    If \( \text{ULNI}(L - r, 4) = 4 \) Then
      If \( \text{ULNI}(L - k, 6) = 0 \) Then
        \( \text{ULNI}(L - k, 6) = \text{ULNI}(L - r, 1) \)
      \end If
      \( \text{ULNI}(L - r, 6) = \text{ULNI}(L - r, 1) \)
      \( \text{ULNI}(L - r, 5) = \text{ULNI}(L - k, 5) \)
      Else
        \( \text{ULNI}(L - r, 6) = \text{ULNI}(L - k, 6) \)
      End If
    End If
  End If
End If

If \( \text{ULNI}(L - r, 4) = 2 \) Then
  If \( \text{ULNI}(L - k, 5) = 0 \) Then
    \( \text{ULNI}(L - k, 5) = \text{ULNI}(L - r, 1) \)
  \end If
  \( \text{ULNI}(L - r, 6) = \text{ULNI}(L - r, 1) \)
  \( \text{ULNI}(L - r, 5) = \text{ULNI}(L - k, 5) \)
  Else
    \( \text{ULNI}(L - r, 5) = \text{ULNI}(L - k, 5) \)
  End If
End If
ULNI(L - r, 7) = L - k
ULNI(L - k, 8) = L - r
ULNI(L - k, 9) = ULNI(L - k, 9) + ULNI(L - r, 9)

End If
End If
Next r

For u = ULNI(L - k, 8) To ULNI(L - k, 7)
If ULNL(u, 2) = ULNL(L - k, 2) And ULNI(u, 3) = g Then
ULNI(u, 5) = ULNI(L - k, 5)
ULNI(u, 6) = ULNI(L - k, 6)
ULNI(u, 8) = ULNI(L - k, 8)
ULNI(u, 9) = ULNI(L - k, 9)
ULNI(u, 11) = ULNI(ULNI(L - k, 8), 1)
ULNI(u, 10) = 1
ULNI(u, 12) = 99
End If
Next u
End If
Next k
Next g

'Clusters < 75 are broken into ULNs

For r = 1 To L
If ULNI(r, 9) < 75 Then
ULNI(r, 5) = 0
ULNI(r, 6) = 0
ULNI(r, 7) = 0
ULNI(r, 8) = 0
ULNI(r, 9) = 0
ULNI(r, 10) = 0
ULNI(r, 11) = 0
ULNI(r, 12) = 0
End If
Next r

'Output ULNs with data

For b = 0 To L
If ULNI(b, 9) >= 100 Then
    v(0) = v(0) + ULNI(b, 0)
End If
Next b

' This sequence will build clusters starting from the end of the TPFDD it names
' clusters using the variable cluster(i,j) starting on RDD cluster(i,0), ending on RDD j,
' and containing cluster(i,1) pax

For p = 1 To 4 '*************comment out to place all in one region************
'p = 1
For g = 1 To 5

For k = 0 To L
If ULNI(L - k, 2) = p And ULNI(L - k, 3) = g And ULNI(L - k, 0) <> 0 And ULNI(L - k, 9) = 0 Then
  ULNI(L - k, 7) = L - k
  ULNI(L - k, 8) = L - k
  ULNI(L - k, 9) = ULNI(L - k, 0)
If ULNI(L - k, 4) = 2 Then
  ULNI(L - k, 5) = ULNI(L - k, 1)
End If
If ULNI(L - k, 4) = 4 Then
  ULNI(L - k, 6) = ULNI(L - k, 1)
End If

For r = k + 1 To L
If r <> k And ULNI(L - r, 2) = p And ULNI(L - r, 3) = g And ULNI(L - r, 7) = 0 Then
  If ULNI(L - k, 6) - ULNI(L - r, 1) <= 4 And ULNI(L - k, 5) - ULNI(L - r, 1) <= 2
  Then
    If ULNI(L - r, 1) = ULNI(ULNI(L - k, 8), 1) And ULNL(L - r, 2) = ULNL(ULNI(L - k, 8), 2)
    Then
      ULNI(L - r, 8) = 9999
    End If
  End If
If ULNI(L - k, 9) < 100 Or ULNI(L - r, 8) = 9999 Then

If ULNI(L - r, 4) = 4 Then
  If ULNI(L - k, 6) = 0 Then
    ULNI(L - k, 6) = ULNI(L - r, 1)
    ULNI(L - r, 6) = ULNI(L - r, 1)
    ULNI(L - r, 5) = ULNI(L - k, 5)
  Else
    ULNI(L - r, 6) = ULNI(L - k, 6)
  End If
End If
End If
If ULNI(L - r, 4) = 2 Then
    If ULNI(L - k, 5) = 0 Then
        ULNI(L - k, 5) = ULNI(L - r, 1)
        ULNI(L - r, 6) = ULNI(L - r, 1)
        ULNI(L - r, 5) = ULNI(L - k, 5)
    Else
        ULNI(L - r, 5) = ULNI(L - k, 5)
    End If
End If

ULNI(L - r, 7) = L - k
ULNI(L - k, 8) = L - r
ULNI(L - k, 9) = ULNI(L - k, 9) + ULNI(L - r, 0)
End If

For u = ULNI(L - k, 8) To ULNI(L - k, 7)
    If ULNI(u, 2) = p And ULNI(u, 3) = g And ULNI(u, 10) = 0 Then
        ULNI(u, 5) = ULNI(L - k, 5)
        ULNI(u, 6) = ULNI(L - k, 6)
        ULNI(u, 8) = ULNI(L - k, 8)
        ULNI(u, 9) = ULNI(L - k, 9)
        ULNI(u, 11) = ULNI(ULNI(L - k, 8), 1)
        ULNI(u, 12) = 98
    End If
Next u
End If
Next k
Next g
Next p

For y = 2 To L
    If ULNI(y, 9) >= 100 Then
        ULNI(y, 10) = 1
    End If
Next y

'output ULNs with their cluster data

For b = 0 To L
If ULNI(b, 9) >= 100 Then
    \( v(1) = v(1) + ULNI(b, 0) \)
End If
Next b

'This sequence tries to combine ULNs in clusters with less than 20 pax with other clusters going to the same POD
'a group of less than 20 pax can be combined with a cluster to the East going to the same POD

For m = 2 To L
    If ULNI(L - m, 0) <> 0 Then
        If ULNI(L - m, 9) < 20 And ULNI(L - m, 12) > 1 Then
            For g = m To L
                If ULNI(L - g, 10) = 1 And g <> m And ULNI(L - g, 12) > 1 And _
                    ULNI(L - m, 2) < ULNI(L - g, 2) And _
                    ULNI(L - m, 8) <> ULNI(L - g, 8) And ULNI(L - m, 11) >= ULNI(L - g, 11) Then
                    If ULNI(L - g, 3) = ULNI(L - m, 3) And _
                        ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And _
                        ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then
                        If ULNI(L - g, 7) > ULNI(L - m, 7) Then
                            LastULN = ULNI(L - g, 7)
                        Else
                            LastULN = ULNI(L - m, 7)
                        End If
                    End If
                End If
            End For
            If ULNI(L - g, 8) < ULNI(L - m, 8) Then
                FirstULN = ULNI(L - g, 8)
            Else
                FirstULN = ULNI(L - m, 8)
            End If
        End If
    End If
End For

totalpax = ULNI(L - g, 9) + ULNI(L - m, 9)

If ULNI(L - g, 5) > ULNI(L - m, 5) Then
    ULNI(L - m, 5) = ULNI(L - g, 5)
Else
    ULNI(L - g, 5) = ULNI(L - m, 5)
End If

If ULNI(L - g, 6) > ULNI(L - m, 6) Then
    ULNI(L - m, 6) = ULNI(L - g, 6)
Else
    ULNI(L - g, 6) = ULNI(L - m, 6)
End If

If ULNI(L - m, 8) = ULNI(L - m, 7) Then
    ULNI(L - m, 12) = m
Else
    For z = ULNI(L - m, 8) To ULNI(L - m, 7)
        If ULNI(z, 7) = ULNI(L - m, 7) Then
            ULNI(z, 12) = m
        End If
    Next z
End If

For z = ULNI(L - g, 8) To ULNI(L - g, 7)
    If ULNI(z, 7) = ULNI(L - g, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For y = FirstULN To LastULN
    If ULNI(y, 12) = m Then
        ULNI(y, 5) = ULNI(L - g, 5)
        ULNI(y, 6) = ULNI(L - g, 6)
        ULNI(y, 7) = LastULN
        ULNI(y, 8) = FirstULN
        ULNI(y, 9) = totalpax
        ULNI(y, 10) = 1
        ULNI(y, 11) = ULNI(ULNI(L - g, 8), 1)
        ULNI(y, 12) = 1
    End If
Next y

    g = L
End If

End If
Next g
End If
End If
Next m

'output ULNs with their cluster data
For b = 0 To L
If ULNI(b, 9) >= 100 Then
    v(2) = v(2) + ULNI(b, 0)
End If
Next b

'This sequence groups clusters with > 40 pax together with clusters from other regions
'which are going to the same POD

For m = 2 To L
    If ULNI(L - m, 9) < 100 And ULNI(L - m, 9) > 40 And ULNI(L - m, 12) > 2 Then
        For g = m To L
            If g <> m And ULNI(L - g, 9) + ULNI(L - m, 9) >= 100 And ULNI(L - g, 9) < 100
                    And ULNI(L - g, 12) > 2 And ULNI(L - m, 8) <> ULNI(L - g, 8) Then
                        If ULNI(L - g, 3) = ULNI(L - m, 3) And _
                            ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And _
                            ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then
                                If ULNI(L - g, 7) > ULNI(L - m, 7) Then
                                    LastULN = ULNI(L - g, 7)
                                Else
                                    LastULN = ULNI(L - m, 7)
                                End If
                        If ULNI(L - g, 8) < ULNI(L - m, 8) Then
                            FirstULN = ULNI(L - g, 8)
                        Else
                            FirstULN = ULNI(L - m, 8)
                        End If
                        ULNI(L - g, 9) = ULNI(L - g, 9) + ULNI(L - m, 9)
                    End If
                End If
        Next g
    End If
Next m
If ULNI(L - g, 6) > ULNI(L - m, 6) Then
    ULNI(L - m, 6) = ULNI(L - g, 6)
Else
    ULNI(L - g, 6) = ULNI(L - m, 6)
End If

For z = ULNI(L - m, 8) To ULNI(L - m, 7)
    If ULNI(z, 7) = ULNI(L - m, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For z = ULNI(L - g, 8) To ULNI(L - g, 7)
    If ULNI(z, 7) = ULNI(L - g, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For y = FirstULN To LastULN
    If ULNI(y, 12) = m Then
        ULNI(y, 5) = ULNI(L - g, 5)
        ULNI(y, 6) = ULNI(L - g, 6)
        ULNI(y, 7) = LastULN
        ULNI(y, 8) = FirstULN
        ULNI(y, 9) = ULNI(L - g, 9)
        ULNI(y, 10) = 1
        ULNI(y, 11) = ULNI(FirstULN, 1)
        ULNI(y, 12) = 2
    End If
Next y

    g = L
    End If
End If
End If

End If
Next g
End If
Next m

'Output Results

For b = 0 To L
    If ULNI(b, 9) >= 100 Then
\[ v(3) = v(3) + \text{ULNI}(b, 0) \]

End If

Next b

' Next Clusters > 40 pax are clustered with clusters > 100 going to the same POD from ' different regions

For m = 2 To L

If \text{ULNI}(L - m, 9) < 100 And \text{ULNI}(L - m, 9) > 40 And \text{ULNI}(L - m, 12) > 3 Then

For g = m To L

If \text{ULNI}(L - g, 10) = 1 And \text{ULNI}(L - g, 9) + \text{ULNI}(L - m, 9) >= 100 And g <> m And \text{ULNI}(L - g, 12) > 3 And \text{ULNI}(L - m, 8) <> \text{ULNI}(L - g, 8) Then

If \text{ULNI}(L - g, 3) = \text{ULNI}(L - m, 3) And

\text{ULNI}(L - m, 6) - \text{ULNI}(\text{ULNI}(L - g, 8), 1) <= 4 And

\text{ULNI}(L - m, 5) - \text{ULNI}(\text{ULNI}(L - g, 8), 1) <= 2 Then

If \text{ULNI}(L - g, 7) > \text{ULNI}(L - m, 7) Then

LastULN = \text{ULNI}(L - g, 7)

Else

LastULN = \text{ULNI}(L - m, 7)

End If

End If

If \text{ULNI}(L - g, 8) < \text{ULNI}(L - m, 8) Then

FirstULN = \text{ULNI}(L - g, 8)

Else

FirstULN = \text{ULNI}(L - m, 8)

End If

\text{ULNI}(L - g, 9) = \text{ULNI}(L - g, 9) + \text{ULNI}(L - m, 9)

End If

If \text{ULNI}(L - g, 5) > \text{ULNI}(L - m, 5) Then

\text{ULNI}(L - m, 5) = \text{ULNI}(L - g, 5)

Else

\text{ULNI}(L - g, 5) = \text{ULNI}(L - m, 5)

End If

If \text{ULNI}(L - g, 6) > \text{ULNI}(L - m, 6) Then

\text{ULNI}(L - m, 6) = \text{ULNI}(L - g, 6)

Else

\text{ULNI}(L - g, 6) = \text{ULNI}(L - m, 6)

End If

61
For $z = \text{ULNI}(L - m, 8)$ to $\text{ULNI}(L - m, 7)$
  If $\text{ULNI}(z, 7) = \text{ULNI}(L - m, 7)$ Then
    $\text{ULNI}(z, 12) = m$
  End If
Next $z$

For $z = \text{ULNI}(L - g, 8)$ to $\text{ULNI}(L - g, 7)$
  If $\text{ULNI}(z, 7) = \text{ULNI}(L - g, 7)$ Then
    $\text{ULNI}(z, 12) = m$
  End If
Next $z$

For $y = \text{FirstULN}$ to $\text{LastULN}$
  If $\text{ULNI}(y, 12) = m$ Then
    $\text{ULNI}(y, 5) = \text{ULNI}(L - g, 5)$
    $\text{ULNI}(y, 6) = \text{ULNI}(L - g, 6)$
    $\text{ULNI}(y, 7) = \text{LastULN}$
    $\text{ULNI}(y, 8) = \text{FirstULN}$
    $\text{ULNI}(y, 9) = \text{ULNI}(L - g, 9)$
    $\text{ULNI}(y, 10) = 1$
    $\text{ULNI}(y, 11) = \text{ULNI}(\text{FirstULN}, 1)$
    $\text{ULNI}(y, 12) = 3$
  End If
Next $y$

$g = L$
End If

End If
Next $g$
End If
Next $m$

'output ULNs with their cluster data
For $b = 0$ to $L$
  If $\text{ULNI}(b, 9) \geq 100$ Then
    $v(4) = v(4) + \text{ULNI}(b, 0)$
  End If
Next $b$

'This sequence groups clusters with > 40 pax together with clusters from the same region
'which are going to a different POD
For m = 2 To L

If ULNI(L - m, 9) < 100 And ULNI(L - m, 9) > 40 And ULNI(L - m, 12) > 4 Then
If ULNI(L - m, 3) = 1 Or ULNI(L - m, 3) = 2 Or ULNI(L - m, 3) = 4 Then
For g = m To L
If g <> m And ULNI(L - g, 9) + ULNI(L - m, 9) >= 100 And ULNI(L - g, 12) > 4 And ULNI(L - m, 12) > 4 And ULNI(L - g, 8) <> ULNI(L - m, 8) Then
If ULNI(L - m, 3) = 1 Or ULNI(L - m, 3) = 2 Or ULNI(L - m, 3) = 4 Then
If ULNI(L - g, 2) = ULNI(L - m, 2) And ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then
If ULNI(L - g, 7) > ULNI(L - m, 7) Then
LastULN = ULNI(L - g, 7)
Else
LastULN = ULNI(L - m, 7)
End If
If ULNI(L - g, 8) < ULNI(L - m, 8) Then
FirstULN = ULNI(L - g, 8)
Else
FirstULN = ULNI(L - m, 8)
End If
End If
ULNI(L - g, 9) = ULNI(L - g, 9) + ULNI(L - m, 9)
End If
End If
If ULNI(L - g, 5) > ULNI(L - m, 5) Then
ULNI(L - m, 5) = ULNI(L - g, 5)
Else
ULNI(L - g, 5) = ULNI(L - m, 5)
End If
If ULNI(L - g, 6) > ULNI(L - m, 6) Then
ULNI(L - m, 6) = ULNI(L - g, 6)
Else
ULNI(L - g, 6) = ULNI(L - m, 6)
End If
For z = ULNI(L - m, 8) To ULNI(L - m, 7)
If ULNI(z, 7) = ULNI(L - m, 7) Then
ULNI(z, 12) = m
End If
Next z

For z = ULNI(L - g, 8) To ULNI(L - g, 7)
    If ULNI(z, 7) = ULNI(L - g, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For y = FirstULN To LastULN
    If ULNI(y, 12) = m Then
        ULNI(y, 5) = ULNI(L - g, 5)
        ULNI(y, 6) = ULNI(L - g, 6)
        ULNI(y, 7) = LastULN
        ULNI(y, 8) = FirstULN
        ULNI(y, 9) = ULNI(L - g, 9)
        ULNI(y, 10) = 1
        ULNI(y, 11) = ULNI(FirstULN, 1)
        ULNI(y, 12) = 4
    End If
Next y

g = L
End If
End If
End If
Next g
End If
End If
Next m

' Next Clusters > 40 pax are clustered with clusters > 100 going from the same region to different PODs

For m = 2 To L
    If ULNI(L - m, 9) < 100 And ULNI(L - m, 9) > 40 And ULNI(L - m, 12) > 5 Then
        If ULNI(L - m, 3) = 1 Or ULNI(L - m, 3) = 2 Or ULNI(L - m, 3) = 4 Then
            For g = m To L
                If ULNI(L - g, 9) + ULNI(L - g, 9) >= 100 And g <> m And ULNI(L - g, 12) > 5 And ULNI(L - m, 8) <> ULNI(L - g, 8) Then
                    If ULNI(L - g, 3) = 1 Or ULNI(L - g, 3) = 2 Or ULNI(L - g, 3) = 4 Then
                        If ULNI(L - g, 2) = ULNI(L - m, 2) And ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And
                    End If
                End If
            Next g
        End If
    End If
Next m
ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then

If ULNI(L - g, 7) > ULNI(L - m, 7) Then
    LastULN = ULNI(L - g, 7)
Else
    LastULN = ULNI(L - m, 7)
End If

If ULNI(L - g, 8) < ULNI(L - m, 8) Then
    FirstULN = ULNI(L - g, 8)
Else
    FirstULN = ULNI(L - m, 8)
End If

ULNI(L - g, 9) = ULNI(L - g, 9) + ULNI(L - m, 9)

If ULNI(L - g, 5) > ULNI(L - m, 5) Then
    ULNI(L - m, 5) = ULNI(L - g, 5)
Else
    ULNI(L - g, 5) = ULNI(L - m, 5)
End If

If ULNI(L - g, 6) > ULNI(L - m, 6) Then
    ULNI(L - m, 6) = ULNI(L - g, 6)
Else
    ULNI(L - g, 6) = ULNI(L - m, 6)
End If

For z = ULNI(L - m, 8) To ULNI(L - m, 7)
    If ULNI(z, 7) = ULNI(L - m, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For z = ULNI(L - g, 8) To ULNI(L - g, 7)
    If ULNI(z, 7) = ULNI(L - g, 7) Then
        ULNI(z, 12) = m
    End If
Next z

For y = FirstULN To LastULN
    If ULNI(y, 12) = m Then
        ULNI(y, 5) = ULNI(L - g, 5)
        ULNI(y, 6) = ULNI(L - g, 6)
        ULNI(y, 7) = LastULN
    End If
Next y
ULNI(y, 8) = FirstULN
ULNI(y, 9) = ULNI(L - g, 9)
ULNI(y, 10) = 1
ULNI(y, 11) = ULNI(FirstULN, 1)
ULNI(y, 12) = 5

End If
Next y

g = L
End If
End If
Next g
Next m

' output ULNs with their cluster data
For b = 0 To L
If ULNI(b, 9) >= 100 Then
    v(5) = v(5) + ULNI(b, 0)
End If
Next b

' Next Clusters > 40 pax are clustered with clusters going from different
' regions to different PODs

For m = 2 To L
    If ULNI(L - m, 9) < 100 And ULNI(L - m, 9) > 40 And ULNI(L - m, 12) > 6 Then
        If ULNI(L - m, 3) = 1 Or ULNI(L - m, 3) = 2 Or ULNI(L - m, 3) = 4 Then
            For g = m To L
                If ULNI(L - g, 9) < 100 And ULNI(L - g, 9) + ULNI(L - m, 9) >= 100 And ULNI(L - g, 12) > 6 And g <> m And ULNI(L - g, 9) >= 40 And ULNI(L - m, 8) <> ULNI(L - g, 8) Then
                    If ULNI(L - g, 3) = 1 Or ULNI(L - g, 3) = 2 Or ULNI(L - g, 3) = 4 Then
                        If ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then
                            If ULNI(L - g, 7) > ULNI(L - m, 7) Then
                                LastULN = ULNI(L - g, 7)
                            Else
                                LastULN = ULNI(L - m, 7)
                            End If
                        End If
                    End If
                End If
            End For
        End If
    End If
Next m
If \( \text{ULNI}(L - g, 8) < \text{ULNI}(L - m, 8) \) Then
   FirstULN = \( \text{ULNI}(L - g, 8) \)
   Else
   FirstULN = \( \text{ULNI}(L - m, 8) \)
End If

\( \text{ULNI}(L - g, 9) = \text{ULNI}(L - g, 9) + \text{ULNI}(L - m, 9) \)

If \( \text{ULNI}(L - g, 5) > \text{ULNI}(L - m, 5) \) Then
   \( \text{ULNI}(L - m, 5) = \text{ULNI}(L - g, 5) \)
   Else
   \( \text{ULNI}(L - g, 5) = \text{ULNI}(L - m, 5) \)
End If

If \( \text{ULNI}(L - g, 6) > \text{ULNI}(L - m, 6) \) Then
   \( \text{ULNI}(L - m, 6) = \text{ULNI}(L - g, 6) \)
   Else
   \( \text{ULNI}(L - g, 6) = \text{ULNI}(L - m, 6) \)
End If

For \( z = \text{ULNI}(L - m, 8) \) To \( \text{ULNI}(L - m, 7) \)
   If \( \text{ULNI}(z, 7) = \text{ULNI}(L - m, 7) \) Then
      \( \text{ULNI}(z, 12) = m \)
   End If
Next z

For \( z = \text{ULNI}(L - g, 8) \) To \( \text{ULNI}(L - g, 7) \)
   If \( \text{ULNI}(z, 7) = \text{ULNI}(L - g, 7) \) Then
      \( \text{ULNI}(z, 12) = m \)
   End If
Next z

For \( y = \text{FirstULN} \) To \( \text{LastULN} \)
   If \( \text{ULNI}(y, 12) = m \) Then
      \( \text{ULNI}(y, 5) = \text{ULNI}(L - g, 5) \)
      \( \text{ULNI}(y, 6) = \text{ULNI}(L - g, 6) \)
      \( \text{ULNI}(y, 7) = \text{LastULN} \)
      \( \text{ULNI}(y, 8) = \text{FirstULN} \)
      \( \text{ULNI}(y, 9) = \text{ULNI}(L - g, 9) \)
      \( \text{ULNI}(y, 10) = 1 \)
      \( \text{ULNI}(y, 11) = \text{ULNI}(\text{FirstULN}, 1) \)
      \( \text{ULNI}(y, 12) = 6 \)
   End If
Next y

    g = L
    End If
    End If
    End If
    Next g
    End If
    End If
    End If
Next m

'output ULNs with their cluster data
For b = 0 To L
If ULNI(b, 9) > 100 Then
    v(6) = v(6) + ULNI(b, 0)
End If
Next b

' Next Clusters > 40 pax are clustered with clusters >100 pax going from different
' regions to different PODs
For m = 2 To L
If ULNI(L - m, 9) < 100 And ULNI(L - m, 9) > 40 And ULNI(L - m, 12) > 7 Then
    If ULNI(L - m, 3) = 1 Or ULNI(L - m, 3) = 2 Or ULNI(L - m, 3) = 4 Then
        For g = m To L
            If ULNI(L - g, 10) = 1 And ULNI(L - g, 12) > 7 And ULNI(L - m, 8) <> ULNI(L - g, 8) Then
                If ULNI(L - g, 3) = 1 Or ULNI(L - g, 3) = 2 Or ULNI(L - g, 3) = 4 Then
                    If ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then
                        If ULNI(L - g, 7) > ULNI(L - m, 7) Then
                            LastULN = ULNI(L - g, 7)
                        Else
                            LastULN = ULNI(L - m, 7)
                        End If
                    End If
                End If
            End If
        End For
    End If
End If

If ULNI(L - g, 8) < ULNI(L - m, 8) Then
    FirstULN = ULNI(L - g, 8)
Else
    FirstULN = ULNI(L - m, 8)
End If
\[ \text{ULNI}(L - g, 9) = \text{ULNI}(L - g, 9) + \text{ULNI}(L - m, 9) \]

If \( \text{ULNI}(L - g, 5) > \text{ULNI}(L - m, 5) \) Then
\[ \text{ULNI}(L - m, 5) = \text{ULNI}(L - g, 5) \]
Else
\[ \text{ULNI}(L - g, 5) = \text{ULNI}(L - m, 5) \]
End If

If \( \text{ULNI}(L - g, 6) > \text{ULNI}(L - m, 6) \) Then
\[ \text{ULNI}(L - m, 6) = \text{ULNI}(L - g, 6) \]
Else
\[ \text{ULNI}(L - g, 6) = \text{ULNI}(L - m, 6) \]
End If

For \( z = \text{ULNI}(L - m, 8) \) To \( \text{ULNI}(L - m, 7) \)
If \( \text{ULNI}(z, 7) = \text{ULNI}(L - m, 7) \) Then
\[ \text{ULNI}(z, 12) = m \]
End If
Next \( z \)

For \( z = \text{ULNI}(L - g, 8) \) To \( \text{ULNI}(L - g, 7) \)
If \( \text{ULNI}(z, 7) = \text{ULNI}(L - g, 7) \) Then
\[ \text{ULNI}(z, 12) = m \]
End If
Next \( z \)

For \( y = \text{FirstULN} \) To \( \text{LastULN} \)
If \( \text{ULNI}(y, 12) = m \) Then
\[ \text{ULNI}(y, 5) = \text{ULNI}(L - g, 5) \]
\[ \text{ULNI}(y, 6) = \text{ULNI}(L - g, 6) \]
\[ \text{ULNI}(y, 7) = \text{LastULN} \]
\[ \text{ULNI}(y, 8) = \text{FirstULN} \]
\[ \text{ULNI}(y, 9) = \text{ULNI}(L - g, 9) \]
\[ \text{ULNI}(y, 10) = 1 \]
\[ \text{ULNI}(y, 11) = \text{ULNI}(\text{FirstULN}, 1) \]
\[ \text{ULNI}(y, 12) = 7 \]
End If
Next \( y \)

\( g = L \)
End If
End If
End If
End If
Next \( g \)
End If
End If
Next m

'output ULNs with their cluster data
For b = 0 To L
If ULNI(b, 9) >= 100 Then
  v(7) = v(7) + ULNI(b, 0)
End If
Next b

'This sequence will look at clusters adjacent to < 100 passenger clusters to see if they can be combined
'The first sequence makes a forward pass through the TPFDD

'All clusters < 90 are broken into single ULN clusters
For r = 1 To L
  If ULNI(r, 9) < 90 And ULNI(r, 10) = 0 Then
    If ULNI(r, 4) = 2 Then
      ULNI(r, 5) = ULNI(r, 1)
      ULNI(r, 6) = 0
    Else
      ULNI(r, 5) = 0
      ULNI(r, 6) = ULNI(r, 1)
    End If
    ULNI(r, 7) = r
    ULNI(r, 8) = r
    ULNI(r, 9) = ULNI(r, 0)
    ULNI(r, 10) = 0
    ULNI(r, 11) = ULNI(r, 1)
  End If
Next r

'This sequence trys to combine unclustered ULNs with other clusters going to the same POD
For m = 2 To L
  If ULNI(L - m, 10) = 0 Then

70
For \( g = m + 1 \) To \( L \)

If \( g <> m \) And ULNI(L - g, 9) + ULNI(L - m, 9) >= 100 Then

If ULNI(L - g, 3) = ULNI(L - m, 3) And ULNI(L - g, 2) >= ULNI(L - m, 2) And ULNI(L - m, 6) - ULNI(ULNI(L - g, 8), 1) <= 4 And ULNI(L - m, 5) - ULNI(ULNI(L - g, 8), 1) <= 2 Then

If ULNI(L - g, 7) > L - m Then

LastULN = ULNI(L - g, 7)

Else

LastULN = L - m

End If

If ULNI(L - g, 8) < L - m Then

FirstULN = ULNI(L - g, 8)

Else

FirstULN = L - m

End If

If ULNI(L - m, 5) = 0 And ULNI(L - m, 6) <> 0 And ULNI(L - m, 6) > ULNI(L - g, 6) Then

ULNI(L - g, 6) = ULNI(L - m, 6)

ElseIf ULNI(L - m, 5) <> 0 And ULNI(L - m, 6) = 0 And ULNI(L - m, 5) > ULNI(L - g, 5) Then

ULNI(L - g, 5) = ULNI(L - m, 5)

End If

ULNI(L - m, 12) = m

For \( z = ULNI(L - g, 8) \) To ULNI(L - g, 7)

If ULNI(z, 7) = ULNI(L - g, 7) Then

ULNI(z, 12) = m

End If

Next \( z \)

ULNI(L - g, 9) = ULNI(L - g, 9) + ULNI(L - m, 0)

For \( y = FirstULN \) To LastULN

If ULNI(y, 12) = m Then

ULNI(y, 5) = ULNI(L - g, 5)

ULNI(y, 6) = ULNI(L - g, 6)

ULNI(y, 7) = LastULN

ULNI(y, 8) = FirstULN
ULNI(y, 9) = ULNI(L - g, 9)
ULNI(y, 10) = 1
ULNI(y, 11) = ULNI(FirstULN, 1)
ULNI(y, 12) = 8

End If
Next y

End If
Next g
End If
Next m

For m = 2 To L

If ULNI(m, 10) = 0 Then
For g = m + 1 To L
  If g <> m And ULNI(g, 9) + ULNI(m, 9) >= 100 Then
    If ULNI(g, 3) = ULNI(m, 3) And ULNI(g, 2) >= ULNI(m, 2) And ULNI(g, 6) - ULNI(m, 1) <= 4 And ULNI(g, 5) - ULNI(m, 1) <= 2 Then
      If ULNI(g, 7) > m Then
        LastULN = ULNI(g, 7)
      Else
        LastULN = m
      End If
      If ULNI(L - m, 5) = 0 And ULNI(L - m, 6) <> 0 And ULNI(L - m, 6) > ULNI(L - g, 6) Then
        ULNI(L - g, 6) = ULNI(L - m, 6)
      ElseIf ULNI(L - m, 5) <> 0 And ULNI(L - m, 6) = 0 And ULNI(L - m, 5) > ULNI(L - g, 5) Then
        ULNI(L - g, 5) = ULNI(L - m, 5)
      End If
    ElseIf ULNI(g, 3) <> 0 And ULNI(g, 2) = 100 And ULNI(g, 7) <= m Then
      If ULNI(g, 5) <> 0 And ULNI(g, 6) = 0 And ULNI(g, 7) > ULNI(m, 7) Then
        ULNI(g, 7) = ULNI(m, 7)
      ElseIf ULNI(g, 5) = 0 And ULNI(g, 6) <> 0 And ULNI(g, 7) > ULNI(m, 7) Then
        ULNI(g, 7) = ULNI(m, 7)
      End If
    End If
  End If
End Next g
End If

Next m
ULNI(m, 12) = m

For z = ULNI(g, 8) To ULNI(g, 7)
    If ULNI(z, 7) = ULNI(g, 7) Then
        ULNI(z, 12) = m
    End If
Next z

ULNI(g, 9) = ULNI(g, 9) + ULNI(m, 0)

For y = FirstULN To LastULN
    If ULNI(y, 12) = m Then
        ULNI(y, 5) = ULNI(g, 5)
        ULNI(y, 6) = ULNI(g, 6)
        ULNI(y, 7) = LastULN
        ULNI(y, 8) = FirstULN
        ULNI(y, 9) = ULNI(g, 9)
        ULNI(y, 10) = 1
        ULNI(y, 11) = ULNI(FirstULN, 1)
        ULNI(y, 12) = 9
    End If
Next y

g = L
End If

End If
Next g
End If
Next m

For b = 0 To L
    If ULNI(b, 9) >= 100 Then
        v(8) = v(8) + ULNI(b, 0)
    End If
Next b

'output ULNs with their cluster data
    t = 1
    For b = 0 To L
        If ULNI(b, 0) <> 0 Then
            Worksheets("ULN").Range("A1").Offset(t, 15 * o + 0) = b
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 1) = ULNL(b, 0)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 2) = ULNL(b, 1)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 3) = ULNI(b, 0)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 4) = ULNI(b, 1)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 5) = ULNI(b, 2)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 6) = ULNI(b, 3)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 7) = ULNI(b, 4)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 8) = ULNI(b, 5)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 9) = ULNI(b, 6)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 10) = ULNI(b, 7)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 11) = ULNI(b, 8)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 12) = ULNI(b, 10)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 13) = ULNI(b, 11)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 14) = ULNI(b, 12)
Worksheets("ULN").Range("A1").Offset(t, 15 * o + 15) = ULNI(b, 12)
t = t + 1
End If
Next b

For k = 0 To 8
    Worksheets("Results").Range("A1").Offset(k, o) = v(k)
Next k

'Next o

End Sub
A MULTI-PASS CONSTRUCTION HEURISTIC FOR THE AGGREGATED AIRLIFT PROBLEM

O’Leary, Stephen T., Captain, USAF

To support the capabilities for precision strike, close air support, intelligence, surveillance, reconnaissance, air mobility and air refueling the Air Force has over 26,000 Airmen deployed from over 80 military installations to Forward Operating Locations (FOLs) in over 21 countries. This research creates a multi-pass construction heuristic to solve the aggregated airlift problem associated with the transportation of these 26,000 deployed Airmen. We are given a Time Phased Force Deployment Document (TPFDD) containing a list of personnel scheduled for deployment, their base of origin, Point of Debarkation (POD), FOL, and Required Delivery Date (RDD). The goal is to aggregate the personnel into groups of 100 or more based on region of origin, POD and RDD to generate dedicated airlift for their deployment. The goal of aggregated airlift is to generate dedicated airlift facilitating better In Transit Visibility of personnel moving through the PODs. This, in turn, enables the proper planning and scheduling of intratheater airlift from the POD to the FOL reducing the backlog of personnel and wait times at the POD. The algorithm was tested on several notional and real world TPFDDs and performed well, increasing the effectiveness over the current manual process by more than 20%.