Shortest Path Across Stochastic Network with Correlated Random Arcs

THESIS

Stephanie M. Boone, 1st Lt, USAF
AFIT-ENC-MS-18-M-109

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

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Army, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.
SHORTEST PATH ACROSS STOCHASTIC NETWORK WITH CORRELATED RANDOM ARCS

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

Stephanie M. Boone, B.S.
1st Lt, USAF

22 March 2018

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
SHORTEST PATH ACROSS STOCHASTIC NETWORK WITH CORRELATED RANDOM ARCS

THESIS

Stephanie M. Boone, B.S.
1st Lt, USAF

Committee Membership:

Lt Col A. J. Geyer
Chair

Dr. R. R. Hill
Reader
Abstract

This paper introduces a new approach to identify the shortest path across a stochastic network with correlated random arcs utilizing nonparametric samples of arc lengths. This approach is applied to find optimal aircraft routes that minimize expected fuel consumption for a given airspeed utilizing predicted wind output from numerical weather prediction (NWP) ensemble models. Results from this new methodology are then compared to the current fuel minimization route planning method that utilizes deterministic NWP wind data for arc lengths. Comparisons are also made to other previously proposed alternative fuel minimization methodologies that utilize mean and median wind data calculated from NWP ensemble wind data.
I would like to thank my husband for his sacrifice and support, without whom this would not be possible. In addition, I would like to thank my faculty research advisor, Lt Col Geyer, and committee member, Dr. Hill, for their guidance, patience, and support in development of this research. I would also like to thank Dr. Baker, Capt Boone, and Col Reiman for their subject matter expertise and guidance in development of this research. Lastly, I would like to thank my classmates for the countless study groups, late nights in the COA, and for making the stressful times more enjoyable.

Stephanie M. Boone
# Table of Contents

Abstract ................................................................. iv  
Acknowledgements ...................................................... v  
List of Figures .......................................................... viii  
List of Tables ........................................................... ix  
List of Acronyms ......................................................... x  
I. Introduction ........................................................... 1  
  1.1 Overview ........................................................ 1  
  1.2 Background ...................................................... 1  
  1.3 Motivation ......................................................... 2  
II. Literature Review ..................................................... 4  
  2.1 Overview ........................................................ 4  
  2.2 Stochastic Shortest Path Problem (SPP) ......................... 4  
    Expected Shortest Path ........................................... 5  
    The Most Shortest Path ........................................... 6  
    The $\alpha$-Shortest Path ......................................... 7  
    Correlated Arcs .................................................. 7  
    The Mean Ensemble Model ....................................... 9  
  2.3 Conclusion ....................................................... 10  
III. Methodology ......................................................... 11  
  3.1 Introduction ..................................................... 11  
    Ensemble NWP .................................................... 11  
    The $a\ postriori$ Shortest Path ................................. 12  
  3.2 AMC Routing Practices ......................................... 12  
    Previous Work .................................................... 12  
    Current Methodology ............................................. 13  
  3.3 The AMC Application ........................................... 14  
    Data Gathering ................................................... 14  
    Building the Network ........................................... 15  
    The Different Models .......................................... 19  
    Model Comparison ............................................... 21  
    Statistical Analysis Techniques ............................... 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV. Analysis</td>
<td>24</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>24</td>
</tr>
<tr>
<td>4.2 KSUU-PHNL</td>
<td>24</td>
</tr>
<tr>
<td>4.3 KTCM-CYQX</td>
<td>26</td>
</tr>
<tr>
<td>4.4 KTCM-KCHS</td>
<td>32</td>
</tr>
<tr>
<td>4.5 KRIV-KWRI</td>
<td>35</td>
</tr>
<tr>
<td>V. Conclusions and Future Research</td>
<td>40</td>
</tr>
<tr>
<td>5.1 Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Future Research</td>
<td>41</td>
</tr>
<tr>
<td>Appendix A. Matlab Code: Data Extraction</td>
<td>42</td>
</tr>
<tr>
<td>A Data Extraction Loop</td>
<td>42</td>
</tr>
<tr>
<td>B Ensemble Data Extraction</td>
<td>44</td>
</tr>
<tr>
<td>C Deterministic Data Extraction</td>
<td>47</td>
</tr>
<tr>
<td>Appendix B. Matlab Code: Network Building</td>
<td>51</td>
</tr>
<tr>
<td>A Linear Interpolation in Time and Space</td>
<td>51</td>
</tr>
<tr>
<td>B Wind Calculations</td>
<td>55</td>
</tr>
<tr>
<td>C Fuel Regression Equations</td>
<td>59</td>
</tr>
<tr>
<td>Appendix C. Matlab Code: The Models</td>
<td>61</td>
</tr>
<tr>
<td>A Deterministic Model</td>
<td>61</td>
</tr>
<tr>
<td>B Homan Model</td>
<td>61</td>
</tr>
<tr>
<td>C IID Model</td>
<td>62</td>
</tr>
<tr>
<td>Bibliography</td>
<td>64</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Outline of Methodology</td>
<td>11</td>
</tr>
<tr>
<td>3.</td>
<td>KSUU-PHNL: RMSE for IID, Deterministic, and Homan Models</td>
<td>25</td>
</tr>
<tr>
<td>4.</td>
<td>KSUU-PHNL: All model comparisons</td>
<td>26</td>
</tr>
<tr>
<td>5.</td>
<td>KSUU-PHNL: True Path Comparisons</td>
<td>27</td>
</tr>
<tr>
<td>6.</td>
<td>KTCM-CYQX: RMSE for IID, Deterministic, and Homan Models</td>
<td>28</td>
</tr>
<tr>
<td>7.</td>
<td>KTCM-CYQX: All model comparisons</td>
<td>29</td>
</tr>
<tr>
<td>8.</td>
<td>KTCM-CYQX: True Path Comparisons</td>
<td>30</td>
</tr>
<tr>
<td>9.</td>
<td>KTCM-KCHS: RMSE for IID, Deterministic, and Homan Models</td>
<td>32</td>
</tr>
<tr>
<td>10.</td>
<td>KTCM-KCHS: All model comparisons</td>
<td>33</td>
</tr>
<tr>
<td>11.</td>
<td>KTCM-KCHS: True Path Comparisons</td>
<td>34</td>
</tr>
<tr>
<td>12.</td>
<td>KRIV-KWRI: RMSE for IID, Deterministic, and Homan Models</td>
<td>36</td>
</tr>
<tr>
<td>13.</td>
<td>KRIV-KWRI: All model comparisons</td>
<td>37</td>
</tr>
<tr>
<td>14.</td>
<td>KRIV-KWRI: True Path Comparisons</td>
<td>38</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Climb $\phi_C$ Regression Terms [42]</td>
<td>18</td>
</tr>
<tr>
<td>2.</td>
<td>Descent $\phi_D$ Regression Terms [42]</td>
<td>18</td>
</tr>
<tr>
<td>3.</td>
<td>Cruise $\phi_D$ Regression Terms [42]</td>
<td>19</td>
</tr>
<tr>
<td>4.</td>
<td>KSUU-PHNL: Unique Route Comparisons</td>
<td>27</td>
</tr>
<tr>
<td>5.</td>
<td>KTCM-CYQX: Unique Route Comparisons</td>
<td>30</td>
</tr>
<tr>
<td>6.</td>
<td>KTCM-KCHS: Unique Route Comparisons</td>
<td>35</td>
</tr>
<tr>
<td>7.</td>
<td>KRIV-KWRI: Unique Route Comparisons</td>
<td>38</td>
</tr>
</tbody>
</table>
List of Acronyms

557th WW    557th Weather Wing
ACFP        Advanced Computer Flight Planner
AFB         Air Force Base
AMC/A3W     AMC Director of Weather
AMC         Air Mobility Command
APSP        \textit{a posteriori} Shortest Path
ATC         Air Traffic Control
CCP         chance-constrained programming
CIF         Cost Index Flying
DCP         dependent-chance programming
DoD         Department of Defense
ft          foot
GFS         Global Forecast System
IID         independently and identically distributed
Klbs        kilopounds
knots       nautical miles per hour
LP          Linear Programming
m/s         meters per second
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAJCOM</td>
<td>Major Command</td>
</tr>
<tr>
<td>MIF</td>
<td>Mission Index Flying</td>
</tr>
<tr>
<td>MSL</td>
<td>mean sea level</td>
</tr>
<tr>
<td>NCEP</td>
<td>National Centers for Environmental Prediction</td>
</tr>
<tr>
<td>NL</td>
<td>Newfoundland and Labrador</td>
</tr>
<tr>
<td>NM</td>
<td>nautical miles</td>
</tr>
<tr>
<td>NOAA</td>
<td>National Oceanic and Atmospheric Administration</td>
</tr>
<tr>
<td>NWP</td>
<td>numerical weather prediction</td>
</tr>
<tr>
<td>RMSE</td>
<td>root mean square error</td>
</tr>
<tr>
<td>SPP</td>
<td>Shortest Path Problem</td>
</tr>
<tr>
<td>TAS</td>
<td>True airspeed</td>
</tr>
<tr>
<td>WARP</td>
<td>Worldwide Aeronautical Flight Planner</td>
</tr>
<tr>
<td>WS</td>
<td>wind speed</td>
</tr>
</tbody>
</table>
I. Introduction

1.1 Overview

With the increased scrutiny on government spending, Air Mobility Command (AMC) has been looking for ways to reduce costs. Fuel has become the largest contributor to aircraft operating costs. As the biggest consumer of aircraft fuel in the Department of Defense (DoD), significant savings could come from more efficient flight planning [23]. According to Lt Col Vince Zabala, AMC’s fuel efficiency program manager, energy costs for the Air Force total nearly $6.8 billion annually, with about 86 percent of that cost spent on aviation fuel [26]. AMC consumes approximately 56 percent, more than all other Major Command (MAJCOM)s combined. If improvements can be made to significantly reduce fuel consumption, AMC could potentially save millions of dollars. In fact, Heseltine [19] determined that $28M a year could be saved if the command saved as little as $200 per sortie.

1.2 Background

The culture in AMC surrounding fuel-efficiency has changed in recent years, but there is still room for improvement. By identifying and utilizing fuel-efficient routes, fuel consumption can be minimized throughout the MAJCOM. While a lot of factors have an impact on fuel efficiency in flight, winds aloft play a large role during long-haul flights. Accurate wind forecasts are vital to ensuring fuel efficiency during
flight planning; Inaccurate forecasts may result in over- or under-estimating the fuel necessary, which translates into wasted money [20].

AMC contractors currently use deterministic numerical weather prediction (NWP) models for aircraft route planning. Deterministic NWP models utilize a single forecast to estimate weather predictions, whereas ensemble models utilize an independently and identically distributed (IID) sample of forecast models to make predictions. These different forecasts (ensemble members) are generated by running multiple simulations with slightly different initial conditions and/or various perturbations of models [36]. The intent is that these model variations represent the range of uncertainty associated with initial weather conditions and yield a range of potential forecasts [14].

Krishnamurti et al. [27] compared ensemble models with their deterministic counterparts and found that the ensemble models illustrated superior forecasting skill over all of the individual models inspected. In the last 30 years, many experts have proposed replacing the traditional deterministic forecast with the ensemble mean forecast [32, 41, 46]. In the last 20 years, ensemble mean forecasts have consistently been found to outperform deterministic forecasts on average [4, 8, 13, 45, 46]. With this consistent improvement upon the traditional deterministic forecasts, the use of ensemble forecasting has become routine [17]. Most recently, Homan [20] used ensemble mean forecasts to predict fuel burn for long range flights and found that ensemble means generally provided more accurate estimates over the deterministic model.

1.3 Motivation

While the use of ensemble NWP data may be routine, it is not common practice in AMC. Therefore, its introduction may provide added value in aircraft routing and fuel estimates. To identify any added value, a technique must be developed that will leverage the uncertainty that is accounted for in the ensemble NWP data.
The stochastic shortest path algorithm is widely used in route planning when there is uncertainty in the model. Unfortunately, however, ensemble NWP output values are highly correlated within ensemble members while randomized but independent between members. Furthermore, the randomized error between ensemble members is nonparametric. These features combine to make creating a stochastic network difficult.

Chapter 2 reviews current methodologies for solving the discrete and probabilistic shortest path problems. In Chapter 3, a new methodology is presented that identifies the shortest path across a stochastic network with correlated random arcs which addresses some limitations of current methodologies. In Chapter 4, this new approach is applied toward the optimal routing of AMC aircraft with respect to minimizing fuel usage and compared to current practices. Finally, Chapter 5 concludes with key insights gained from this research and propose efforts to further this research, as well as additional applications of this new methodology.
II. Literature Review

2.1 Overview

This chapter discusses the stochastic Shortest Path Problem (SPP) and three current methodologies for determining the shortest path: the expected shortest path, the most shortest path, and the $\alpha$-shortest path. The goal of the classic (deterministic) SPP is to find the quickest, cheapest, and/or the most reliable route between two points [1]. These problems are very common when dealing with transportation, routing, and communication networks. However, the discrete SPP is not always the most realistic, particularly when the arc lengths are uncertain. For instance, the optimal routing of an aircraft between two points will be highly affected by winds; predictions of which are highly probabilistic in ensemble NWP models. In situations such as these where there is significant uncertainty in the network, the classic SPP is far from sufficient [48]. The robust, or stochastic, SPP varies from its classic counterpart wherein the length of each arc is associated with a probability distribution [6]. Several different models have been proposed when solving this type of problem. Three of these models are the expected shortest path, the most shortest path, and the $\alpha$-shortest path. For comparison, the Linear Programming (LP) formulation for the deterministic SPP is shown in (1).

2.2 Stochastic SPP

The three stochastic shortest path models discussed below are variations of their deterministic counterpart. The objective function varies with each model, but the constraints from the deterministic model remain constant in each variation, as does its notations. $A$ is the set of all arcs $(i, j)$ in the network and $x_{ij}$ defines the arc from node $i$ to node $j$, where $1 \leq i, j \leq n$. If the arc $(i, j)$ is in the path, then $x_{ij}=1$ and
0 otherwise. $c_{ij}$ is the cost of traversing, or the length of, the arc $(i, j)$.

$$
\min \sum_{(i,j) \in A} c_{ij}x_{ij}
$$
subject to:

$$
\sum_{(1,j) \in A} x_{1j} - \sum_{(j,1) \in A} x_{j1} = 1, \\
\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = 0, \ 2 \leq i \leq n - 1, \\
\sum_{(n,j) \in A} x_{nj} - \sum_{(j,n) \in A} x_{jn} = -1, \\
x_{ij} \in 0, 1, \ \forall (i, j) \in A.
$$

**Expected Shortest Path**

The expected shortest path finds the path with the shortest expected length between two nodes; that is, the path that is shortest on average [43]. Murthy and Sarkar [34] found that finding the expected shortest path reduces to the discrete SPP where the arc costs are replaced by their expected values. There has been extensive research in the development of formulas to calculate these solutions. Davis and Prieditis [11] developed a closed-form approximation, building upon the recursive method developed by Kulkarni [28]. They determined the expected shortest path when arcs are independent and exponentially distributed. Davis and Prieditis [11] also found that their same formula gives a close approximation when the arcs are uniformly distributed. Ji [24] identified a general linear formulation for identifying the expected shortest path as (2).

The only difference between the expected shortest path formulation (2) and the deterministic case (1), is the objective function. Instead of minimizing the cost to get from the source node to the terminus node, the goal is to minimize the expected cost and identify the path that is shortest on average. $E\left[\sum_{(i,j) \in A} \xi_{ij}x_{ij}\right]$ is the expected shortest path and $\xi_{ij}$ is the arc length, with the associated probability distribution,
from nodes 1 to \( n \).

\[
\begin{align*}
\min & \quad E \left[ \sum_{(i,j) \in A} \xi_{ij} x_{ij} \right] \\
\text{subject to :} & \\
\sum_{(1,j) \in A} x_{1j} - \sum_{(j,1) \in A} x_{j1} &= 1, \\
\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} &= 0, \quad 2 \leq i \leq n - 1, \\
\sum_{(n,j) \in A} x_{nj} - \sum_{(j,n) \in A} x_{jn} &= -1, \\
x_{ij} &\in 0, 1, \quad \forall (i, j) \in A.
\end{align*}
\tag{2}
\]

The Most Shortest Path

The most shortest path model determines the path that has the highest probability of being faster than some requirement \( T_0 \) \cite{24}. Using the dependent-chance programming (DCP) concepts outlined by Liu \cite{29}, Ji \cite{24} developed the following DCP model for the most shortest path shown in (3). The most shortest path formulation contains the same constraints as the deterministic model, again the only variability is in the objective function.

\[
\begin{align*}
\max & \quad Pr \left\{ \sum_{(i,j) \in A} \xi_{ij} x_{ij} \leq T_0 \right\} \\
\text{subject to :} & \\
\sum_{(1,j) \in A} x_{1j} - \sum_{(j,1) \in A} x_{j1} &= 1, \\
\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} &= 0, \quad 2 \leq i \leq n - 1, \\
\sum_{(n,j) \in A} x_{nj} - \sum_{(j,n) \in A} x_{jn} &= -1, \\
x_{ij} &\in 0, 1, \quad \forall (i, j) \in A.
\end{align*}
\tag{3}
\]
The $\alpha$-Shortest Path

The $\alpha$-shortest path identifies the path that minimizes some time constraint $\bar{T}$ with a confidence level of at least $\alpha$ [24]. Leveraging the chance-constrained programming (CCP) concepts developed by Charnes and Cooper [10] and Liu [30], Ji [24] developed a model for determining the $\alpha$-shortest path shown in (4). The deterministic formulation in (1) is augmented with an additional constraint and new objective function.

$$\min \bar{T}$$
subject to :

$$\Pr\left\{ \sum_{(i,j)\in A} \xi_{ij}x_{ij} \leq \bar{T} \right\} \geq \alpha,$$

$$\sum_{(1,j)\in A} X_{1j} - \sum_{(j,1)\in A} X_{j1} = 1,$$

$$\sum_{(i,j)\in A} X_{ij} - \sum_{(j,i)\in A} X_{ji} = 0, \quad 2 \leq i \leq n - 1,$$

$$\sum_{(n,j)\in A} X_{nj} - \sum_{(j,n)\in A} X_{jn} = -1,$$

$$X_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A.$$

These and many other methodologies assume arc lengths to be independently distributed to simplify models and reduce computational complexity [22]. While this assumption may be necessary to simplify a problem and its computational complexity, it is extremely limiting to ignore such a strong characteristic of the network and dampens the strength of the result. For this reason, newer methodologies have been introduced that account for arc correlation.

Correlated Arcs

There are many instances where arc lengths are not only uncertain, but they are correlated. For example, groups of nodes or links in a specific region of the net-
work may be correlated and, in turn, adjacent links and nodes are also affected [15].
Therefore, when determining the shortest path, prior choices will inform the deci-
sion making for the duration of the path construction, i.e. identifying the shortest
path. Fan et al. [15] outline a formulation for solving the shortest path problem with
correlated arcs. The arcs are considered congested or not congested, affected or not
affected. Conditional probabilities are associated with each arc; that is, the proba-
bility that node $i$ is affected given that node $i - 1$ is affected. Fan et al. [15] present
two formulations to identify the expected shortest path between two nodes depend-
ing on the type of network: node-based or link-based congestion. These networks
are described by where the congestion may occur, at the nodes as in the node-based
approach or along the arcs as in the link-based approach.

Eq. (5) is the formulation when node-based congestion is present. Where $u_{ij} =$
the lowest expected travel time from uncongested node $i$ to node $j$ and $v_{ij} =$ the
lowest expected travel time from congested node $i$ to $j$, where $i = 1, 2, ..., N - 1$
and $j = 2, 3, ..., N$. The $\alpha_{ij}$ is the probability that if node $i$ is uncongested, then
node $j$ is uncongested and $\beta_{ij}$ is the probability that if node $i$ is congested then $j$ is
congested. The $t_{ij}$ and $\tau_{ij}$ are the expected arc lengths from $(i, j)$ under uncongested
and congested conditions respectively.

\[
\begin{align*}
    u_i &= \min_{j \neq i} \{ t_{ij} + \alpha_{ij}u_j + (1 - \alpha_{ij})v_j \}, \quad i = 1, 2, ..., N - 1 \\
    v_i &= \min_{j \neq i} \{ \tau_{ij} + \beta_{ij}v_j + (1 - \beta_{ij})u_j \}, \quad i = 1, 2, ..., N - 1 \\
    u_N, v_N &= 0
\end{align*}
\]  

(5)

Eq. (6) shows the formulation when link-based congestion is present. The same
notation from the node-based formulation also apply here. However, there are addi-
tional variables that need to be defined. That is,
\[ \lambda_{ij} = 1 - \beta_{ij}, \]

\[ p_{ij}(\tau)d\tau = \text{the probability that traveling } (i, j) \text{ requires time between } \tau \text{ and } \tau + d\tau \]
given that the arc traversed to node \( i \) was \textit{uncongested}, and

\[ q_{ij}(\tau)d\tau = \text{the probability that traveling } (i, j) \text{ requires time between } \tau \text{ and } \tau + d\tau \]
given that the arc traversed to node \( i \) was \textit{congested}.

\[
\begin{align*}
    u_i &= \min_j \{ t_{ij} + \alpha_{ij}u_j + (1 - \alpha_{ij})v_j \}, \quad i = 1, 2, ..., N - 1 \\
    v_i &= \min \{ \tau_{ij} + \lambda_{ij}u_j + (1 - \lambda_{ij})v_j \}, \quad i = 1, 2, ..., N - 1 \\
    u_N, v_N &= 0
\end{align*}
\]  

(6)

where:

\[ t_{ij} = \int_0^\infty \tau p_{ij}(\tau)d\tau \text{ and} \]

\[ \tau_{ij} = \int_0^\infty \tau q_{ij}(\tau)d\tau. \]

While the formulations proposed address correlated arcs, they are contingent on conditional probabilities, which are difficult for the AMC problem. In addition, Fan \textit{et al.} [15] requires assumptions about the distribution of probabilities across each arc which is not possible in the AMC problem.

**The Mean Ensemble Model**

Homan [20] compared the fuel burn estimates of a mean ensemble model to those of the deterministic approach currently in use today. Specifically, the study compared the fuel loads planned using a deterministic model forecast to those using three different ensemble mean forecasts across five aircraft and five pre-determined routes. The +00 hour forecast was used as the ‘truth’ source for a given date/time for the previous forecasts at the same date/time. The ‘true’ fuel burn was compared to the estimates
for the previous forecasts to calculate a fuel burn error. The results suggested that
the use of ensemble means generally provided more accurate estimates.

While this approach might provide a better fuel point estimate than the deter-
ministic approach currently in use, there is still a great deal of data that is not
being utilized. By averaging the ensembles, data that could provide additional in-
sight toward fuel planning is lost. More accurately, winds at point A and time 1 are
correlated to winds at point B upstream at time 0. Correlation information within
each ensemble member is lost, therefore averaging across the ensembles may not be
the best approach.

2.3 Conclusion

While these examples are just a subset of the many ways to approach the stochastic
SPP, many of these approaches only work when the arcs are independent. Of the
formulations that allow for correlation, there are other limitations that do not fully
address the AMC problem. A new methodology is outlined in the next chapter that
determines the shortest path across a correlated random network without assuming
independence of arc lengths or a distribution for the arc costs.
III. Methodology

3.1 Introduction

This chapter outlines the methodology for the *a posteriori* Shortest Path (APSP) approach proposed in this research and for the case study provided. The AMC application is described in detail to show how the data were gathered, how the network was constructed, and how this model differs from previous approaches. Lastly, information is provided showing which statistical techniques were applied to the results to gain further insight. Figure 1 provides an overview of this methodology.

```
for each route do
| Extract Weather Data every 6 hours for 7 days using |
| GFS Deterministic Model, and |
| Ensemble NWP Model |
| Build Network |
| Create Nodes |
| interpolate data across time and space |
| Calculate Arc Costs |
| \( c_{ij} \) = fuel required to fly from Node \( i \) to Node \( j \) |
| leverage data and fuel equations to calculate each \( c_{ij} \) |
| Determine Shortest Path using |
| Deterministic Model, |
| Human Model, and |
| IID Model |
| Analyze |
| Compare all Model Results |
| Using IID Model Results, calculate |
| Mean and Median Confidence Intervals |
| \( t \)-tests on multiple optimal routes, if applicable |
end
```

*Figure 1. Outline of Methodology*

**Ensemble NWP**

NWP leverages current weather observations and computer models to forecast future weather [38]. Current AMC models utilize NWP deterministic forecasts as opposed to the readily available ensemble forecasts. A deterministic forecast is a single member of an ensemble model initialized without random perturbations [16].
In other words, this type of model utilizes one forecast, whereas ensemble forecasts utilize multiple forecasts for weather prediction.

**The a posteriori Shortest Path**

Unlike current methodologies where the arcs are probabilistic, the stochastic nature of this network is analyzed after-the-fact. Using structural factoring, the complex network is broken down into $k$ subnetworks [18]. Each subnetwork is then solved as the discrete SPP, formulation in (1), to obtain optimal solution(s) [6]. These $k$ optimal solutions are then analyzed using nonparametric statistics, *i.e.* kernel smoothing, to obtain descriptive statistics and other relevant analyses on the $k$ solutions.

### 3.2 AMC Routing Practices

AMC is the largest single consumer of fuel in the DoD. As such, their focus has shifted toward a more fuel-efficient culture and a great deal of work has been done to identify more fuel-efficient practices.

**Previous Work**

Mirtich [33] introduced the concept of Cost Index Flying (CIF). This is a program now used by commercial airlines to balance the cost of time and the cost of fuel. The USAF has since adopted this program and renamed it Mission Index Flying (MIF). Weather data is leveraged when determining aircraft routing, and in the last few years, research has improved the current routing practices. Homan [20] compared ensemble mean and deterministic forecasts for route planning. Homan’s results suggested that ensemble mean forecasts outperform deterministic forecasts. That is, ensemble mean forecasts provide more accurate fuel burn estimates which could result in less reserve fuel being carried. However, unlike Mirtich, Homan’s research has yet to be adopted.
Current Methodology

This new APSP methodology arose in an effort to provide better fuel estimates to AMC. AMC currently utilizes the Advanced Computer Flight Planner (ACFP) system to route cargo and ensure aerial refueling operations [40]. This is done by optimizing routes with respect to fuel consumption, subject to aircraft performance with wind and temperatures aloft and air traffic control and diplomatic constraints [20]. The 557th Weather Wing (557th WW) provides ACFP with the weather data necessary for this optimization scheme.

Weather data are extracted from a single forecast, one-degree NWP model at six-hour increments, from six to 96 hours, for each waypoint along the route [3]. These data consist of wind data for each latitude/longitude pair at each of the 4 atmospheric pressure levels [39]. Note that additional weather data are available but were not used in this analysis.

Wind data are presented with U- and V- components in meters per second (m/s). These components are the East/West and North/South components of the wind, respectively. Positive U-component indicate that the wind is traveling West to East. Positive V-component indicates that the wind is traveling from South to North. These components are utilized to determine the wind speed, direction, and angle.

Atmospheric Pressure Levels are provided in millibars. These pressure levels, when combined with temperature at a given point, translate to the altitude above mean sea level (MSL).

At the core of ACFP is the Worldwide Aeronautical Flight Planner (WARP). WARP serves to leverage advanced search techniques to produce routes that minimize fuel burn [40]. According to the AMC Director of Weather (AMC/A3W), routes are broken down into segments, legs, and sublegs within WARP (Fig. 2) [3]. Segments lie between two points. That is, if a route has four points, then that route would have
three segments. Legs lie between two navigational points along the route. Legs are then divided into sublegs inside WARP. If the length of a cruise leg is greater than 60 miles, then WARP divides the leg into sublegs such that all sublegs are less than or equal to 60 miles. During climb, sublegs are divided into 4000-foot (ft) increments. Weather at each subleg is determined at the midpoint of that subleg and the average of these subleg midpoints within each leg is reported as the weather for that leg. [3]

To get the weather data for a specific point (latitude, longitude, altitude, and time), WARP interpolates the weather data from nearby, known points, specifically from lower altitude and earlier time to higher altitude and later time. For example, if a specific point is 75% through a time slot, then the temperature and winds will be representative of 75% of the change in the temperature or wind, respectively.

3.3 The AMC Application

Data Gathering

Two MATLAB scripts were written to extract NWP data directly from National Oceanic and Atmospheric Administration (NOAA) using nctoolbox developed by Schlining et al. [44] [21]. Wind data, temperature, pressure levels, and model step times are provided at each latitudinal and longitudinal coordinate. Given the latitudinal and longitudinal coordinates, the U- and V- wind components are extracted in
6-hr increments across 20 ensemble members. Given the latitudinal and longitudinal coordinates, the U- and V- wind components are extracted in 3-hr increments across 1 deterministic forecast.

**Building the Network**

To investigate the problem at hand, three models were developed using different NWP data sets: a deterministic model, IID ensemble model, and Homan (mean ensemble) model. Each of these shortest path models are identical, the only difference being how the initial weather conditions are input into the model.

The weather data for the deterministic model are provided with one forecast in 3-hr increments with 1-degree resolution. These weather data from NOAA are extracted for three pressure altitudes converted to altitude MSL in standard atmosphere: 25K, 30K, and 35K feet. Operators will not change altitude in 5K-ft increments, therefore linear interpolation over time and space is used to calculate weather data in 1K-feet increments between 25K and 35K feet and hourly increments of time.

The weather data for the IID and Homan models are provided with 20 ensemble members in 6-hr increments with 1-degree resolution. Therefore, the network is separated into 20 subnetworks, where subnetwork \(i\) leverages the weather data at ensemble \(i\). These weather data from NOAA are extracted at the same three pressure levels as in the deterministic forecast and interpolated in the same manner. To validate model comparisons, described in detail later in Section 3.3, the weather data for all models begins with the +06 hour forecast.

Each subnetwork consists of \(11n + 1\) nodes \((i)\) and \(11(11n - 20)\) arcs \((j)\), where \(n\) is the number of equidistant legs along the great circle route from the source node \(s\) to the terminus node \(t\). The weather data are provided from a 1 degree Global Forecast System (GFS) model, therefore a weighted average based on the location of
actual node with respect to the nearest 1 degree nodes was applied to determine the weather data at each node.

Ng et al. [37] found that travel time and fuel savings for initial climb and final descent are negligible when compared to those during cruise. For this reason, the take-off and landing portions of flight are ignored; that is, node 1 and node 11n + 1 are forced to be 25k feet.

According to the C-17 Fact Sheet, the average cruising speed is 450 nautical miles per hour (knots) [12]. Therefore, this research assumes that the C-17 maintains a constant airspeed of 450 knots during cruise. To calculate the effect that winds have on fuel efficiency, that is, the headwind, tailwind, and crosswind, the wind speed and direction and the aircraft heading is first calculated. Weather data are provided in U- and V- components in m/s, therefore the wind speed (m/s) was calculated using the Pythagorean Theorem as (7).

\[ WS = \sqrt{U^2 + V^2} \]  \hspace{1cm} (7)

The wind direction was calculated using MATLAB’s \texttt{atan2d} function, (8), to calculate the wind direction in degrees. MATLAB’s \texttt{atan2d}(y,x) function returns the four-quadrant arctangent of \( y/x \) [31].

\[ \text{Wind Direction} = \text{atan2d}(-U, -V) \]  \hspace{1cm} (8)

The aircraft heading from A to B, without any wind effects, was calculated using the \texttt{atan2d} function as (9) [47].

\[ \text{AC heading}_{i-1} = \text{mod(atan2d}(Y, X), 360) \]  \hspace{1cm} (9)
where:

\[ X = \cos \theta_A \sin \theta_B - \sin \theta_A \cos \theta_B \cos \Delta L, \]
\[ Y = \cos \theta_B \sin \Delta L, \]
\[ L = \text{Longitude}, \]
\[ \theta = \text{Latitude}. \]

The aircraft corrected heading, due to winds, was calculated using MATLAB’s \texttt{driftcorr} function which takes AC\_heading, True airspeed (TAS), Wind\_Dir, and wind speed (WS) as inputs and returns the aircraft’s corrected heading (AC\_heading\_corr), ground speed (in knots), and the correction angle (in degrees) due to winds. The distance between neighboring nodes were small enough to be assumed linear; therefore Pythagorean Theorem was again used to determine the straight-line distance between nodes.

Reiman [42] developed regression models on flight data from performance manuals to estimate fuel consumption for the C-17, C-130, and C-5. These models were utilized to determine the path that required the least amount of fuel to traverse. These models were broken down into climb, cruise, and descent. For the purposes of this problem, only the C-17 data are provided in the tables.

The regression model for calculating the fuel and distance required to climb in (10) and their respective \( \beta \)s are shown in Table 1. The descent model for calculating the the fuel and distance required for descent in (11) with the \( \beta \)s in Table 2.

\[
\phi_C = \beta_0 + \beta_1 \alpha + \beta_2 \alpha^2 + \beta_3 \alpha^3 + \beta_4 \omega + \beta_5 \omega^2 + \beta_6 \omega^3 + 10^{-6} \beta_7 \alpha^2 \omega^3 + 10^{-6} \beta_8 \alpha^2 \omega^3 \quad (10)
\]

\[
\phi_D = \beta_0 + \beta_1 \omega + \beta_2 \omega^2 + \beta_3 \alpha + \beta_4 \alpha \omega \quad (11)
\]
where:

\( \phi_C \) = Fuel to Climb in Klbs or Distance to Climb in NMs
\( \phi_D \) = Fuel to Descend in Klbs or Distance to Descend in NMs
\( \alpha \) = Altitude in Thousands of Feet
\( \omega \) = Aircraft Gross Weight in Klbs at Climb/Descent Start

**Table 1. Climb \( \phi_C \) Regression Terms [42]**

<table>
<thead>
<tr>
<th>( \beta ) term</th>
<th>Fuel</th>
<th>Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>-4.7054</td>
<td>-51.504</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.2869</td>
<td>2.0961</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>-0.0070</td>
<td>-0.0282</td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>7.1E-05</td>
<td>0.0003</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>0.0267</td>
<td>0.3363</td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>-5.9E-05</td>
<td>-0.0008</td>
</tr>
<tr>
<td>( \beta_6 )</td>
<td>4.8E-08</td>
<td>6.9E-07</td>
</tr>
<tr>
<td>( \beta_7 )</td>
<td>6.7E-05</td>
<td>0.0003</td>
</tr>
<tr>
<td>( \beta_8 )</td>
<td>-2.1E-07</td>
<td>1.7E-05</td>
</tr>
</tbody>
</table>

**Table 2. Descent \( \phi_D \) Regression Terms [42]**

<table>
<thead>
<tr>
<th>( \beta ) term</th>
<th>Fuel</th>
<th>Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.2574</td>
<td>-16.382</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.0005</td>
<td>0.1278</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>-8.5E-7</td>
<td>-1.7E-4</td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>0.0108</td>
<td>1.3919</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>3.2E-5</td>
<td>0.0036</td>
</tr>
</tbody>
</table>

Finally, the cruise model for calculating the fuel consumed during cruise in (12) with the respective \( \beta \) values in Table 3.

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

\[
- \frac{1}{3A} \sqrt[3]{\frac{1}{2} \left[ 2B^3 - 9ABC + 27A^2D - \sqrt{(2B^3 - 9ABC + 27A^2D)^2 - 4(B^2 - 3AC)^3} \right]}
\]

where (all weights in Klbs):

\[
A = \frac{\beta_4}{3}
\]

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

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

\[
D = -\delta
\]

\( \alpha \) = Altitude in Thousands of Feet
\( \delta = \text{Distance in NMs} \)
\( \omega = \text{Aircraft Gross Weight} \)
\( \omega_{frc} = \text{Reserve/Contingency Fuel Weight} \)
\( \omega_{op} = \text{Operating Weight} \)
\( \omega_{fah} = \text{Alternate/Holding Fuel Weight} \)
\( \omega_p = \text{Payload Weight} \)
\( \omega_{ff} = \text{Cruise Fuel Weight} \)
\( f = \text{Fuel Consumed} \)
\[
= \omega_{op} + \omega_{frc} + \omega_{fah} + \omega_p + f
\]

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>31.735</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.9897</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>-0.0043</td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>-0.0642</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>5.8E-05</td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>-0.0011</td>
</tr>
</tbody>
</table>

A binary integer programming model is formulated to determine the shortest path, with respect to fuel consumed, between two points. This model utilizes the great circle route between the two points, and optimizes the cruising altitude to minimize fuel consumption along that route. This model is the deterministic model in (1) where \( c_{ij} \) is calculated utilizing the regression models developed by Reiman [42], code is provided in Appendix B and C.

The Different Models

This subsection identifies the differences between the three models developed for this analysis. Each model utilizes the same network(s), however the weather data is implemented into each model differently. Only one approach, the IID Model, actually uses the new \textit{a posteriori} methodology.
Each of the models are applied to four operationally relevant routes as identified by a subject matter expert, a C-17 Instructor Pilot [7]. The following routes are used:

KSUU-PHNL Travis Air Force Base (AFB), CA to Honolulu, HI
KTCM-CYQX McChord AFB, WA to Gander Newfoundland and Labrador (NL)
KTCM-KCHS McChord AFB, WA to Charleston AFB, SC
KRIV-KWRI March Air Reserve Base to Joint Base McGuire-Dix-Lakehurst, NJ

**Deterministic Model**

The purpose of the deterministic model was to replicate current AMC routing practices. AMC currently uses deterministic forecasts for aircraft routing, therefore a single forecast was used to determine the optimal path: NOAA’s deterministic 1°GFS model forecast. These weather data were utilized to identify the route that would minimize fuel burn; see Appendix A for the MATLAB code for the deterministic model. This network does not utilize the *a posteriori* approach proposed because there is no uncertainty accounted for in the model. Therefore, no subnetworks are analyzed.

**Homan Model**

The Homan model is a recreation of the mean ensemble model introduced by Homan [20]. This model is similar to the deterministic model in that it does not utilize the *a posteriori* approach developed and reverts to the discrete SPP. However, unlike the deterministic model, the Homan model leverages the ensemble data. Instead of using one deterministic forecast, the average of all of the ensembles is input into the network for the mean model, code is provided in Appendix B. However, there are
issues with this model due to the inter-correlation between wind values within each ensemble member.

**IID Model**

The IID model leverages the *a posteriori* approach. Twenty subnetworks are developed, one for each ensemble. The optimal path for each subnetwork is then saved. With these 20 optimal paths, up to 20 unique routes are identified. Each unique route is then re-ran through each of the subnetworks again, code provided in Appendix C. This provided 20 estimates of fuel consumption for each unique route. Descriptive statistics were then applied to determine which routes were statistically significantly better (consume less fuel) than others across all ensembles and to develop confidence intervals on the fuel estimates.

**Model Comparison**

The accuracy of each of the three models is calculated using the root mean square error (RMSE) of the fuel burn estimates. The model estimates are compared to a truth value in order to calculate the fuel burn error. This truth is calculated using the Deterministic Model across the +00 hour forecasts. The +00 hour forecast for a specific date/time is the initialization of the deterministic model and is therefore the closest thing to the true conditions at each GFS model run time. This method of comparison was used by Homan [20] and is a common technique for NWP researchers [25]. The RMSE is calculated as (13) where $N$ is the sample size, $FB_{truth}$ is the true fuel burn, and $FB_{est}$ is the estimated fuel burn for a given model. $N = 1$ for the Deterministic and Homan models and $N = 20$ for the IID model.

$$FB_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (FB_{est} - FB_{truth})_i^2}$$ (13)
Statistical Analysis Techniques

The aforementioned descriptive statistics allow a better characterization of fuel usage across flights. Through $t$-tests, any statistically significant differences between the optimal route for each subnetwork can be identified. This will identify the overall optimal route(s) and, if multiple routes are identified, potentially provide the user with route options that consume statistically equivalent amounts of fuel.

Using the estimated fuel consumption for each ensemble will also result in fuel usage confidence intervals. These intervals provide the user with more fidelity when calculating the amount of fuel required for a mission. The ensemble data can provide the user with the $(1 - \alpha)\%$ confidence interval around the mean fuel usage. This is shown in (14) where $\mu$ is the true mean, $\bar{x}$ is the sample mean, $n$ is the number of ensembles (20), and $\sigma$ is the sample standard deviation.

$$
\mu \in \bar{x} \pm t_{1-\frac{\alpha}{2},n-1}\frac{\sigma}{\sqrt{n}}
$$

Confidence intervals around the median fuel usage can also be calculated by determining the values of $j$ and $k$ such that $P(X_{(j)} \leq x_p \leq X_{(k)}) = 1 - \alpha$ after sorting the data from smallest to largest [9]. This is shown in (15) where $n$ is the number of ensembles (20) and $q$ is the proportion (0.5). Therefore, the 95% confidence interval of the median when $n = 20$ is always between $X_6$ and $X_{15}$.

$$
j = \left\lceil nq - t_{1-\frac{\alpha}{2},n-1}\sqrt{nq(1-q)} \right\rceil
$$

$$
k = \left\lfloor nq + t_{1-\frac{\alpha}{2},n-1}\sqrt{nq(1-q)} \right\rfloor
$$

Comparing the different models and the unique routes identified in the IID model is done using $t$-tests, more specifically the paired $t$-test. Because fuel estimates depend on the specific forecasts used and are not independent, the paired $t$-test is the most appropriate test for detecting statistically significant differences in the fuel estimates.
The paired $t$-test tests the null hypothesis, $H_0 : \mu_2 - \mu_1 = d_0$, versus the alternative, $H_1 : \mu_2 - \mu_1 \neq d_0$, where $d_0$ is the difference to detect and $\mu_i$ is the true mean of group $i$. For this study, $d_0 = 0$ because the goal of the test is to identify if there is a difference between the true means. Let $\bar{d}$ and $s_d$ be the sample mean and standard deviation of the differences, respectively, and $n$ be the number of observations, then the critical value, $t_0$ is calculated as (16) [5]. The statistic, $t_0$, is then tested against the test statistic to determine if there is a statistically significant difference between the means. If $|t_0| \geq t_{1-\alpha/2,n-1}$, then there is enough evidence to identify a statistically significant difference between the means with $(1 - \alpha)$% confidence.

$$t_0 = \frac{\bar{d} - d_0}{s_d \sqrt{n}}$$  \hspace{1cm} (16)
IV. Analysis

4.1 Introduction

For the deterministic and mean models, only one route and fuel point estimate is calculated. However, with the IID model, 20 different estimates of fuel usage and at least one route is identified. To show the differences between the models, seven consecutive days of weather data, ranging from 28 January to 5 February 2018, were extracted for each route of interest every six hours. This resulted in 27 deterministic and ensemble weather forecasts for each of the routes.

4.2 KSUU-PHNL

The RMSE of fuel burn estimates for the KSUU-PHNL route for the IID, Deterministic, and Homan models are shown in Figure 3. The RMSE for the Homan and IID models are nearly identical, whereas the RMSE using the Deterministic model is much larger in all but two cases.

The Homan model point estimates are well contained in both mean and median confidence intervals generated by the IID model, as seen in Figure 4. However, the fuel estimate yielded by the deterministic model is outside of the 95% mean confidence interval 27/27 times. When outside the confidence bounds, the deterministic model under- or over-estimates between 21.85 and 4,414.44 pounds of fuel at each time interval. The deterministic model under-estimated and over-estimated 27 times. Figure 4 shows the offsets between the current deterministic model with all other models, including the 95% mean and median confidence intervals calculated using the IID model.

The large spread between the deterministic model and Homan model is verified by the paired $t$-tests. These tests show that there is a statistically significant difference
between the deterministic and Homan models, with a p-value of $3.69 \times 10^{-8}$.

Aside from the ability to generate confidence intervals around the mean and median, there is another advantage to the IID model. Of the 27 timesteps, 13 found multiple routes across all ensembles. These timesteps and their results are shown in Table 4. In 11 scenarios, only two unique routes were identified, and three routes were identified in the other two scenarios. 13 of the 17 total route comparisons performed identified a statistically significant difference between the means. So, depending on the day and time, multiple alternative routes could be suggested that will not use statistically significant more fuel.

Figure 3. KSUU-PHNL: RMSE for IID, Deterministic, and Homan Models
For each date/time, the optimal routes identified by each model were compared to the optimal path of the *true* forecast. In 10 of 25 comparisons, the IID model identified the *true* optimal path (Figure 5). The Deterministic and Homan models did not identify the *true* path in any of the scenarios inspected. Figure 3 shows which date/times each model identified the *true* optimal path.

4.3 KTCM-CYQX

The RMSE results for the KTCM-CYQX route for the IID, Deterministic, and Homan models are shown in Figure 6. The RMSE for the Homan and IID models are
Table 4. KSUU-PHNL: Unique Route Comparisons

<table>
<thead>
<tr>
<th>Date &amp; Time</th>
<th>Number of Routes</th>
<th>Comparison</th>
<th>p-value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 Jan 0600</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.02 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$2.1 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td>31 Jan 1200</td>
<td>3</td>
<td>$\mu_2 = \mu_3$</td>
<td>$5.67 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_2$</td>
<td>0.038</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$6.72 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$3.85 \times 10^{-8}$</td>
<td>≠</td>
</tr>
<tr>
<td>31 Jan 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.151</td>
<td>=</td>
</tr>
<tr>
<td>2 Feb 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$5.00 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>2 Feb 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.052</td>
<td>=</td>
</tr>
<tr>
<td>2 Feb 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.117</td>
<td>=</td>
</tr>
<tr>
<td>3 Feb 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.45 \times 10^{-10}$</td>
<td>≠</td>
</tr>
<tr>
<td>3 Feb 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.021</td>
<td>=</td>
</tr>
<tr>
<td>4 Feb 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.67 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>4 Feb 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.140</td>
<td>=</td>
</tr>
<tr>
<td>4 Feb 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$4.03 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td>5 Feb 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.015</td>
<td>≠</td>
</tr>
<tr>
<td>5 Feb 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$6.91 \times 10^{-8}$</td>
<td>≠</td>
</tr>
</tbody>
</table>
nearly identical, whereas the RMSE using the Deterministic model is much larger in all but three cases.

Figure 6. KTCM-CYQX: RMSE for IID, Deterministic, and Homan Models

The Homan model point estimates are well contained in the mean and median confidence intervals generated by the IID model, as seen in Figure 7. However, the fuel estimate yielded by the deterministic model is outside of the 95% mean confidence interval 24/24 times. When outside the confidence bounds, the deterministic model under- or over-estimates between 186.25 and 2,731.89 pounds of fuel at each time interval. The deterministic model under-estimated and over-estimated 24 times. Figure 7 shows the offsets between the current deterministic model with all other
models, including the 95% mean and median confidence intervals calculated using the IID model.

The large spread between the deterministic model and Homan model is verified by the paired $t$-tests. These tests show that there is a statistically significant difference between the deterministic and Homan models, with a p-value of $8.01 \times 10^{-9}$.

Aside from the ability to generate confidence intervals around the mean and median, there is another advantage to the IID model. Of the 27 timesteps, 14 found multiple routes across all ensembles. These timesteps and their results are shown in Table 4. In six scenarios, only two unique routes were identified and the number of

![Figure 7. KTCM-CYQX: All model comparisons](image-url)

The large spread between the deterministic model and Homan model is verified by the paired $t$-tests. These tests show that there is a statistically significant difference between the deterministic and Homan models, with a p-value of $8.01 \times 10^{-9}$.

Aside from the ability to generate confidence intervals around the mean and median, there is another advantage to the IID model. Of the 27 timesteps, 14 found multiple routes across all ensembles. These timesteps and their results are shown in Table 4. In six scenarios, only two unique routes were identified and the number of
routes identified in the other eight scenarios ranged from three to seven. A total of
29 of the 47 total route comparisons performed identified a statistically significant
difference between the means. Depending on the day and time, multiple alternative
routes could be suggested that will not use statistically significant more fuel.

For each date/time, the optimal routes identified by each model were compared
to the optimal path of the true forecast. In 16 of 25 comparisons, the IID model
identified the true optimal path (Figure 8). The Deterministic and Homan models
did not identify the true path in any of the scenarios inspected. Figure 6 shows which
date/times each model identified the true optimal path.

![Figure 8. KTCM-CYQX: True Path Comparisons](image)

**Table 5. KTCM-CYQX: Unique Route Comparisons**

<table>
<thead>
<tr>
<th>Date &amp; Time</th>
<th>Number of Routes</th>
<th>Comparison</th>
<th>p-value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 Jan 1800</td>
<td>4</td>
<td>$\mu_1 = \mu_2$</td>
<td>$3.83 \times 10^{-5}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$1.84 \times 10^{-9}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_4$</td>
<td>$3.35 \times 10^{-8}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$7.99 \times 10^{-10}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_4$</td>
<td>$2.66 \times 10^{-8}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_4$</td>
<td>$3.36 \times 10^{-7}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>Date &amp; Time</td>
<td>Number of Routes</td>
<td>Comparison</td>
<td>p-value</td>
<td>Result</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
<td>------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>30 Jan 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.87 \times 10^{-7}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>30 Jan 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.057</td>
<td>$=$</td>
</tr>
<tr>
<td>30 Jan 1200</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$6.60 \times 10^{-7}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$2.11 \times 10^{-6}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$1.03 \times 10^{-3}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>30 Jan 1800</td>
<td>7</td>
<td>$\mu_1 = \mu_2$</td>
<td>$5.65 \times 10^{-2}$</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>0.124</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_4$</td>
<td>0.602</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_5$</td>
<td>0.996</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_6$</td>
<td>0.027</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_7$</td>
<td>0.022</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_8$</td>
<td>0.337</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_5$</td>
<td>0.638</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_6$</td>
<td>0.027</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_4$</td>
<td>0.318</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_5$</td>
<td>0.602</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_7$</td>
<td>0.028</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_7$</td>
<td>0.023</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_5$</td>
<td>0.124</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_6$</td>
<td>0.019</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_7$</td>
<td>0.015</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_5 = \mu_6$</td>
<td>0.023</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_5 = \mu_7$</td>
<td>0.019</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_6 = \mu_7$</td>
<td>0.124</td>
<td>$=$</td>
</tr>
<tr>
<td>31 Jan 0000</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.046</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>0.027</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>0.083</td>
<td>$=$</td>
</tr>
<tr>
<td>31 Jan 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.93 \times 10^{-6}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>31 Jan 1200</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$8.37 \times 10^{-5}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$3.86 \times 10^{-6}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$3.37 \times 10^{-6}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>31 Jan 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$6.33 \times 10^{-6}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>1 Feb 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.527</td>
<td>$=$</td>
</tr>
<tr>
<td>1 Feb 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.10 \times 10^{-3}$</td>
<td>$\neq$</td>
</tr>
<tr>
<td>2 Feb 1800</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.401</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>0.458</td>
<td>$=$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>0.671</td>
<td>$=$</td>
</tr>
<tr>
<td>3 Feb 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.079</td>
<td>$=$</td>
</tr>
<tr>
<td>4 Feb 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$3.56 \times 10^{-8}$</td>
<td>$\neq$</td>
</tr>
</tbody>
</table>
4.4 KTCM-KCHS

The RMSE results for the KSUU-PHNL route for the IID, Deterministic, and Homan models are shown in Figure 9. The RMSE for the Homan and IID models are nearly identical, whereas the RMSE using the Deterministic model is much larger in all scenarios.

The Homan model point estimates are well contained in both mean and median confidence intervals generated by the IID model, as seen in Figure 10. However, the fuel estimate yielded by the deterministic model is outside of the 95% mean confidence interval 25/25 times. The deterministic model over-estimates the true fuel estimate.
between 2,981.81 and 6,940.05 pounds of fuel at each time interval. Figure 10 shows the offsets between the current deterministic model with all other models, including the 95% mean and median confidence intervals calculated using the IID model. The Homan model overlaps the mean confidence interval in every scenario.

The large spread between the deterministic model and Homan model is verified by the paired $t$-tests. These tests show that there is a statistically significant difference between the deterministic and Homan models, with a p-value of $5.01 \times 10^{-21}$.

Aside from the ability to generate confidence intervals around the mean and median, there is another advantage to the IID model. Of the 26 timesteps, 13 found
multiple routes across all ensembles. These timesteps and their results are shown in Table 6. In 11 scenarios, only two unique routes were identified, and three routes were identified in the other two scenarios. All 17 total route comparisons performed identified a statistically significant difference between the means. Depending on the day and time, multiple alternative routes could be suggested that will not use statistically significant more fuel.

For each date/time, the optimal routes identified by each model were compared to the optimal path of the true forecast. In 15 of 25 comparisons, the IID model identified the true optimal path (Figure 11). The Deterministic and Homan models did not identify the true path in any of the scenarios inspected. Figure 9 shows which date/times each model identified the true optimal path.

![Count of Times True Path Identified or Missed](image)

**Figure 11. KTCM-KCHS: True Path Comparisons**
### Table 6. KTCM-KCHS: Unique Route Comparisons

<table>
<thead>
<tr>
<th>Date &amp; Time</th>
<th>Number of Routes</th>
<th>Comparison</th>
<th>P value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 Jan 0000</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.36 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$1.10 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$7.27 \times 10^{-7}$</td>
<td>≠</td>
</tr>
<tr>
<td>28 Jan 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.48 \times 10^{-3}$</td>
<td>≠</td>
</tr>
<tr>
<td>28 Jan 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.03 \times 10^{-2}$</td>
<td>≠</td>
</tr>
<tr>
<td>28 Jan 1800</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$4.46 \times 10^{-3}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$7.0 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$7.8 \times 10^{-7}$</td>
<td>≠</td>
</tr>
<tr>
<td>29 Jan 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.89 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>29 Jan 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.17 \times 10^{-2}$</td>
<td>≠</td>
</tr>
<tr>
<td>29 Jan 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.59 \times 10^{-7}$</td>
<td>≠</td>
</tr>
<tr>
<td>29 Jan 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$5.5 \times 10^{-8}$</td>
<td>≠</td>
</tr>
<tr>
<td>30 Jan 0600</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$5.0 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td>31 Jan 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.11 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>31 Jan 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$6.45 \times 10^{-6}$</td>
<td>≠</td>
</tr>
<tr>
<td>4 Feb 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.52 \times 10^{-5}$</td>
<td>≠</td>
</tr>
</tbody>
</table>

### 4.5 KRIV-KWRI

The RMSE for the KRIV-KWRI route for the IID, Deterministic, and Homan models are shown in Figure 12. The RMSE for the Homan and IID models are nearly identical, and the RMSE of the Deterministic model varies with the other models.

The Homan model point estimates are well contained in both mean and median confidence intervals generated by the IID model, as seen in Figure 13. However, the fuel estimate yielded by the deterministic model is outside of the 95% mean confidence interval 24 of 26 times. The deterministic model under-estimates the true fuel burn between 235 and 1,707.09 pounds of fuel at each time interval. The deterministic model under-estimated 27 times. Figure 13 shows the offsets between the current deterministic model with all other models, including the 95% mean and median confidence intervals calculated using the IID model.
The large spread between the deterministic model and Homan model is verified by the paired $t$-tests. These tests show that there is a statistically significant difference between the deterministic and Homan models, with a p-value of $2.35 \times 10^{-2}$.

Aside from the ability to generate confidence intervals around the mean and median, there is another advantage to the IID model. Of the 24 timesteps, 10 found multiple routes across all ensembles. These timesteps and their results are shown in Table 7. In five scenarios, only two unique routes were identified and the number of routes identified in the other five scenarios ranged from three to six. A total of 34 of the 42 total route comparisons performed identified a statistically significant dif-
ference between the means. So, depending on the day and time, multiple alternative routes could be suggested that will not use statistically significant more fuel.

For each date/time, the optimal routes identified by each model were compared to the optimal path of the true forecast. In 3/24 comparisons, the IID model identified the true optimal path (Figure 14). The Deterministic and Homan models did not identify the true path in any of the scenarios inspected. Figure 12 shows which date/times each model identified the true optimal path.

Figure 13. KRIV-KWRI: All model comparisons
Figure 14. KRIV-KWRI: True Path Comparisons

Table 7. KRIV-KWRI: Unique Route Comparisons

<table>
<thead>
<tr>
<th>Date &amp; Time</th>
<th>Number of Routes</th>
<th>Comparison</th>
<th>p-value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 Jan 0600</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$3.26 \times 10^{-2}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>0.236</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$9.90 \times 10^{-6}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td>31 Jan 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$3.69 \times 10^{-4}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td>31 Jan 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>0.269</td>
<td></td>
</tr>
<tr>
<td>1 Feb 1800</td>
<td>5</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.81 \times 10^{-8}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$5.88 \times 10^{-5}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_4$</td>
<td>0.035</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_5$</td>
<td>0.236</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>0.299</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_4$</td>
<td>0.148</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_5$</td>
<td>0.034</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_4$</td>
<td>$3.22 \times 10^{-4}$</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_5$</td>
<td>0.020</td>
<td>(\neq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_5$</td>
<td>0.038</td>
<td>(\neq)</td>
</tr>
<tr>
<td>Date &amp; Time</td>
<td>Number of Routes</td>
<td>Comparison</td>
<td>p-value</td>
<td>Result</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>------------</td>
<td>-----------------------</td>
<td>--------</td>
</tr>
<tr>
<td>1 Feb 0000</td>
<td>6</td>
<td>$\mu_1 = \mu_2$</td>
<td>$4.10 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$7.87 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_4$</td>
<td>$2.00 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_5$</td>
<td>$1.81 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_6$</td>
<td>$2.02 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$6.07 \times 10^{-5}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_4$</td>
<td>$1.95 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_5$</td>
<td>$1.76 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_6$</td>
<td>$1.97 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_4$</td>
<td>$2.75 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_5$</td>
<td>$2.46 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_6$</td>
<td>$2.78 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_5$</td>
<td>$6.90 \times 10^{-9}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_4 = \mu_6$</td>
<td>$0.013$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_5 = \mu_6$</td>
<td>$0.595$</td>
<td>=</td>
</tr>
<tr>
<td>2 Feb 0000</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$1.24 \times 10^{-9}$</td>
<td>≠</td>
</tr>
<tr>
<td>3 Feb 1800</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$2.69 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>4 Feb 0600</td>
<td>3</td>
<td>$\mu_1 = \mu_2$</td>
<td>$4.66 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$0.011$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$1.01 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td>4 Feb 1200</td>
<td>2</td>
<td>$\mu_1 = \mu_2$</td>
<td>$0.101$</td>
<td>=</td>
</tr>
<tr>
<td>4 Feb 1800</td>
<td>4</td>
<td>$\mu_1 = \mu_2$</td>
<td>$3.13 \times 10^{-7}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_3$</td>
<td>$5.27 \times 10^{-7}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_1 = \mu_4$</td>
<td>$6.51 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_3$</td>
<td>$0.949$</td>
<td>=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_2 = \mu_4$</td>
<td>$5.08 \times 10^{-4}$</td>
<td>≠</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\mu_3 = \mu_4$</td>
<td>$2.34 \times 10^{-4}$</td>
<td>≠</td>
</tr>
</tbody>
</table>
V. Conclusions and Future Research

5.1 Conclusion

By incorporating ensemble NWP into the route planning, AMC can reduce the amount of excess fuel burned by poor forecasts. The Homan model performed well when compared to the IID but due to the inability to account for the inter-correlation within each ensemble member and the correlation across ensembles, this model is not ideal.

Of the three models discussed, the Deterministic model almost always over-estimated fuel burn compared to the IID and Homan models; sometimes up to almost 4,500 pounds of fuel. For one of the four routes inspected, the Deterministic model under-estimated the fuel burn up to 2,000 pounds. While these over-estimations of fuel consumption could result in up to 671 gallons of excess fuel, these estimates are only for a small subset of coast-to-coast routes that AMC flies regularly and only for the C-17 aircraft. When not over-estimating the fuel necessary, the Deterministic model under-estimated the fuel required up to almost 300 pounds of fuel. The average RMSE for the IID and Homan models across all routes investigated was roughly 512 and 501 pounds, respectively, while the average RMSE of the Deterministic model was almost 2,500 pounds of fuel. On average, the Deterministic model misses the true fuel burn by nearly 2,000 more pounds than the IID and Homan models. These severe inconsistencies in fuel burn estimates can make it difficult for appropriate route planning. The amount of wasted or insufficient fuel will add up quickly for longer, i.e. transoceanic, routes resulting in excess costs or dangerous situations.

Applying the APSP methodology in the IID model provides users with additional information and more fidelity. The deterministic and Homan models provide point estimates. The IID model provides a range of potential values which further aids in
flight planning. It has the ability to provide multiple routes that will not statistically change the amount of fuel used. Additionally, the IID model was the only model to ever identify the true optimal path during the testing period.

5.2 Future Research

This research only accounts for aircraft performance and weather in determining the optimal route. A more useful flight plan should also include route restrictions from Air Traffic Control (ATC) and relevant regulatory restrictions [2]. This application only accounts for the effects of wind on fuel consumption, what about other weather conditions? As a proof of concept, several simplifying assumptions were made: constant TAS, linearity between time, space, and points, and only looking at the great circle route. Building upon these assumptions would yield a more accurate tool for identifying optimal aircraft routing and estimating fuel consumption. Another interesting expansion of this application would be to look at lateral route deviations instead of just vertical deviations.

This research focused on shorter, coast-to-coast routes. While differences between the current methodology and the application of the APSP, a study should be conducted to investigate the differences between the models for longer flights. In addition, a study should also be conducted to identify a distribution of fuel by month, day, etc. This study could leverage the +00 hour forecasts from this analysis and the National Centers for Environmental Prediction (NCEP) GFS Historical Archive [35], which has weather data from 15 January 2015 to 21 February 2018.

AMC aircraft are large and can handle the affects of wind better than smaller aircraft. This technique could provide valuable insight to the small aircraft and drone communities, because they are heavily impacted by winds. This a posteriori approach could also apply to communication networks.
Appendix A. Matlab Code: Data Extraction

A Data Extraction Loop

Code/PullData.m

1 clear;
2 clc;
3
4 % Add the current directory and subdirectories to the path
5 addpath(genpath(pwd()));
6
7 % Get the current UTC
8 t1 = datetime('now','TimeZone','utc');
9
10 % Strip off the minutes and seconds
11 t1 = t1 - minutes(minute(t1)) - seconds(second(t1));
12
13 % Count back to the model time
14 while(mod(hour(t1),6) ~= 0)
15     t1 = t1 - hours(1);
16 end
17
18 t2 = t1 - days(7); % go back 7 days
19
20 str_trial = 100;
21 num_trial = 1;
22 stepsize = 100;
23 nlegs = 100;
24
25 % Pressure levels
26 levels = [400,350,300,250]; % pressure levels interested in
27
28 num_mem = 20; % number of ensemble members
29 TAS = 450; % constant TAS
30
31
32 routes = {'KSUU-PHNL','KTCM-CYQX','KTCM-KCHS','KRIV-KWRI'};
33
34 % lat/longs from airnav.com
35
36 latlongs_routes = [38.2645167, -121.9241315, 21.3178275, -157.9202627; % Travis lat/long, Honolulu
47.1376778, 32.8986389, -80.0405278; % McChord Lat/Long, Gander
47.1376778, -122.4764750, 48.936944, -54.567778; % Travis lat/long
33.8819433, -117.2590169, 40.0155833, -74.5916991]; % March Air Reserve Air
40 base, JBMDL lat/long

41 [num_routes,~] = size(latlongs_routes);
while (t1 >= t2) % Loop back seven days

for i = 1:num_routes % create all folders based on route names before pulling data

    FolderName = sprintf(’%s’, routes{i}); % save in folder named after route
    mkdir(’./Thesis Docs/Data’, FolderName) % create new folder under data tab

    LatA = latlongs_routes(i,1);
    LongA = latlongs_routes(i,2);
    LatB = latlongs_routes(i,3);
    LongB = latlongs_routes(i,4);
    latlongmat = gcwaypts(LatA,LongA,LatB,LongB,nlegs);

    % Pull Weather Data

    % Hours in forecast
    tot_dist = deg2nm(distance(’gc’,[LatA,LongA],[LatB,LongB])); % calculate total distance
    from pt A to pt B across great circle route in nautical miles
    totalHrs = 2*ceil(tot_dist/TAS); % double time to travel across gc route and round up (in hours) -&gt; amount of hours at least to pull weather data for

    while (mod(totalHrs,6) ~= 0) % Adjust the total hours to a multiple of 6 (model is in 6 hour increments)
        totalHrs = totalHrs + 1;
    end

    % Left/Right longitude
    leftlon = mod(floor(min(latlongmat(:,2))),360);
    rightlon = mod(ceil(max(latlongmat(:,2))),360);

    % Top/Bottom latitude

    toplat = ceil(max(latlongmat(:,1))); % ceil(LatA);
    bottomlat = floor(min(latlongmat(:,1))); % floor(LatB);

    err = 1;
    while err == 1

        % Gets the winds and temps
        [~, U, V, W, lat, lon, iso] = Ensemble_Wind_Temp( t1, levels, ...
                totalHrs, leftlon, rightlon, toplat, bottomlat );

        if (isempty(U) || isempty(V) || isempty(W) || ...
                isempty(lat) || isempty(lon) || isempty(iso))
            disp(’Route %s, download failed for ensemble GFS run starting %s, %s’);
            err = 1;
            %return;
            else

                %file_name =
                sprintf(’%s’, routes{i}, ’ensemble.mat’, day(W(1)), month(W(1)), hour(W(1)), 0);

                if month(W(1)) == 2
                    mnth = ’Feb’;
                else mnth = ’Jan’;
                end

                file_name = sprintf(’%s’, routes{i}, ’ensemble.mat’, day(W(1)), mnth, hour(W(1)), 0)
    
    end

end

end
matfile = fullfile(pwd, FolderName, file_name);
save(matfile, 'U', 'V', 'lat', 'lon', 'iso');
err = 0;
end

while err == 0
    [~, U, V, W, lat, lon, iso] = Deterministic_Wind_Temp(t1, levels, ...
    totalHrs, leftlon, rightlon, toplat, bottomlat);
    if (isempty(U) || isempty(V) || isempty(W) || ...
        isempty(lat) || isempty(lon) || isempty(iso))
        disp(strcat('Route ', routes{i}, ' download failed for deterministic GFS run starting ', datestr(t1)));
        err = 0;
        %return;
    else
        file_name = sprintf('%02i%02i%02i_Deterministic.mat', day(W(1)), month(W(1)), hour(W(1)), 0);
        file_name = sprintf('%02i%02i%02i_Deterministic.mat', day(W(1)), mnth, hour(W(1)), 0);
        matfile = fullfile(pwd, FolderName, file_name);
        save(matfile, 'U', 'V', 'lat', 'lon', 'iso');
        err = 1;
    end
end

end

t1 = t1 - hours(6);  % decrement model initialization time by 6 hours

B Ensemble Data Extraction

function [ T, U, V, W, lat, lon, iso ] = Ensemble_Wind_Temp(t1, levels, ...
    totalHrs, leftlon, rightlon, toplat, bottomlat )

% This function retrieves the GFS 1 degree ensemble Numerical Weather
% Prediction (NWP) model wind and temperature output for the requested
% level(s) and time period(s).

% This function requires the netoolbox package:
% https://github.com/netoolbox/netoolbox
% Make sure to include the netoolbox directory in the MATLAB path.

% Input variables:
% levels - isobaric pressure levels (1000, 925, 850, 700, 500, 400 and/or 300 mb)
% totalHrs - total number of hours of forecast data needed
% leftlon - Western edge of longitude window (0 - 360)
% rightlon - Eastern edge of longitude window (0 - 360)
% toplat - Northern edge of latitude window (-90 - 90)
% bottomlat - Southern edge of latitude window (-90 - 90)
W = [];  % NWP model step times
lat = [];  % Latitudes in degrees (-90 - 90)
lon = [];  % Longitudes in degrees (0 - 360)
iso = [];  % Isobaric pressure levels (translates to altitude MSL)

% Approximate altitude calculations from pressure can be found here:
% http://ww2010.atmos.wisc.edu/gh/wwlpr/constant_pressure_surface.rxml

% Dimensions in order are time (entries in W), ensemble member (1 to 20),
% pressure level (entries in iso in millibars), latitude, and longitude
T = [];  % Temperature values in degrees Celsius
U = [];  % U component of winds in meters per second
V = [];  % V component of winds in meters per second

% Explanation for how to use U and V wind components can be found here:
% http://colaweb.gmu.edu/dev/clim301/lectures/wind/wind-uv.html

% Make sure top and bottom lat are correctly configured
if bottomlat > toplat
    temp = bottomlat;
    bottomlat = toplat;
    toplat = temp;
end

if toplat > 90 || bottomlat < -90
    return;
end

% Make sure left and right longitudes are correctly configured
if leftlon > rightlon
    temp = rightlon;
    rightlon = leftlon;
    leftlon = temp;
end

if leftlon < 0 || rightlon > 360
    return;
end

% Set up the pressure levels to ensure only standard levels are entered
levels = intersect(levels,[1000,975,950,925,900,850,800,750,700,650,600,...
550,500,475,450,400,350,300,250,200,150,125,100,70,50,30,20,10,7,5,2,1]);

% Set up nctoolbox
setup_nctoolbox;

% Count back to the model time
while (mod(hour(t1),6) ~= 0)
    t1 = t1 - hours(1);
end

% Adjust the total hours to a multiple of 6 (model is in 6 hour increments)
while (mod(totalHrs,6) ~= 0)
    totalHrs = totalHrs + 1;
end
\( t_2 = t_1 - \text{hours}(12) \);

\( \text{DownLoadError} = 1; \)

\( \textbf{while} (\text{DownLoadError} && t_1 >= t_2) \)

\( \% \text{ Store model data times} \)

\( W = t_1 + \text{hours}(0:6:totalHrs); \)

\( \textbf{for} j = 0:6:totalHrs \)

\( \textbf{for} i = 1:20 \)

\( \% \text{The weather data URL} \)

\( \text{URL} = \text{'http://nomads.ncep.noaa.gov/cgi-bin/filter_gens.pl?file=gpe'}; \)

\( \text{URL} = \text{strcat(URL, num2str(i, '%02i'), '.t', num2str(hour(t1), '%02i'))}; \)

\( \text{URL} = \text{strcat(URL, 'z_pgrb2f', num2str(j, '%02i'))}; \)

\( \text{for k = 1:max(size(levels))} \)

\( \text{URL} = \text{strcat(URL, '&lev=', num2str(levels(k), 'mb=on'))}; \)

\( \text{end} \)

\( \text{URL} = \text{strcat(URL, '&var_TMP=on&var_UGRD=on&var_VGRD=on&subregion=&leftlon=', num2str(leftlon))}; \)

\( \text{URL} = \text{strcat(URL, 'rightlon=', num2str(rightlon), '&toplat=', num2str(toplat), ...} \)

\( \text{&bottomlat=', num2str(bottomlat), '&dir=%2Fgifs.'}); \)

\( \text{URL} = \text{strcat(URL, num2str(year(t1))}); \)

\( \text{URL} = \text{strcat(URL, num2str(month(t1), '%02i'))}; \)

\( \text{URL} = \text{strcat(URL, num2str(day(t1), '%02i'))}; \)

\( \text{URL} = \text{strcat(URL, num2str(hour(t1), '%02i'), '%2Fpgrb2')}; \)

\( \text{fileName} = \text{strcat(pwd(), '/winds%', num2str(j), '.', num2str(i), '.grib2')}; \)

\( \% \text{Download weather data} \)

\( \text{try} \)

\( \text{outfilename = webservice(fileName, URL)}; \)

\( \text{catch} \)

\( \text{if (j==0 && i == 1)} \)

\( t_1 = t_1 - \text{hours}(6); \)

\( \text{totalHrs = totalHrs + 6; } \)

\( \text{DownLoadError} = 1; \)

\( \text{break; } \)

\( \text{else} \)

\( \text{return; } \)

\( \text{end} \)

\( \text{end} \)

\( \% \text{Import weather data into MATLAB} \)

\( \text{nc = ncgeodataset(outfilename)}; \)

\( \% \text{Fill values for T, U and V} \)

\( V(floor(j/6)+1,:), :, :, :) = \text{nc('v-component_of_wind_isobaric')}; \)

\( U(floor(j/6)+1,:), :, :, :) = \text{nc('u-component_of_wind_isobaric')}; \)

\( T(floor(j/6)+1,:), :, :, :) = \text{nc('Temperature_isobaric')}; \)

\( \text{DownLoadError} = 0; \)

\( \text{end} \)

\( \text{if (DownLoadError == 1)} \)

\( \text{break;} \)
if (DownLoadError == 1)
    return;
end

lat = nc('lat')(:);
lon = nc('lon')(:);
iso = nc('isobaric')(:)/100;

for j = 0:6:totalHrs
    for i = 1:20
        outfilename = strcat(pwd(),'/winds_',num2str(j),'.',num2str(i),'.grib2');
        % Delete the weather data file
        delete(outfilename);
        delete(strcat(outfilename,'.gbx9'));
        delete(strcat(outfilename,'.ncx'));
    end
end

C Deterministic Data Extraction

Code/Deterministic_Wind_Temp.m

function [ T, U, V, W, lat, lon, iso ] = Deterministic_Wind_Temp( t1, levels, ...
    totalHrs, leftlon, rightlon, toplat, bottomlat )
% This function retrieves the GFS 1 degree deterministic Numerical Weather
% Prediction (NWP) model wind and temperature output for the requested
% level(s) and time period(s).

% This function requires the netoolbox package:
% https://github.com/netoolbox/netoolbox
% Make sure to include the netoolbox directory in the MATLAB path.

% Input variables:
% levels - isobaric pressure levels (1000,925,850,700,500,400 and/or 300 mb)
% totalHrs - total number of hours of forecast data needed
% leftlon - Western edge of longitude window (0 - 360)
% rightlon - Eastern edge of longitude window (0 - 360)
% toplat - Northern edge of latitude window (-90 - 90)
% bottomlat - Southern edge of latitude window (-90 - 90)

W = []; % NWP model step times
lat = []; % Latitudes in degrees (-90 - 90)
lon = []; % Longitudes in degrees (0 - 360)
iso = []; % Isobaric pressure levels (translates to altitude MSL)
% Approximate altitude calculations from pressure can be found here:
% http://ww2010.atmos.uiuc.edu/(Gh)/wwslpr/constant_pressure_surface.xml
% Dimensions in order are time (entries in W), ensemble member (1 to 20),
% pressure level (entries in iso in millibars), latitude, and longitude
T = []; % Temperature values in degrees Celsius
U = []; % U component of winds in meters per second
V = []; % V component of winds in meters per second

% Explanation for how to use U and V wind components can be found here:
% http://colaweb.gmu.edu/dev/clim301/lectures/wind/wind-wv.html

% Make sure top and bottom lat are correctly configured
if bottomlat > toplat
    temp = bottomlat;
    bottomlat = toplat;
    toplat = temp;
end

if toplat > 90 || bottomlat < -90
    return;
end

% Make sure left and right longitudes are correctly configured
if leftlon > rightlon
    temp = rightlon;
    rightlon = leftlon;
    leftlon = temp;
end

if leftlon < 0 || rightlon > 360
    return;
end

% Set up the pressure levels to ensure only standard levels are entered
levels = intersect(levels,[1000,975,950,925,900,850,800,750,700,650,600,...
                         550,500,475,450,400,350,300,250,200,150,125,100,70,50,30,20,10,7,5,2,1]);

% Set up nctoolbox
setup_nctoolbox;

% Count back to the model time
while (mod(hour(t1),6) ~= 0)
    t1 = t1 - hours(1);
end

% Adjust the total hours to a multiple of 6 (model is in 6 hour increments)
while (mod(totalHrs,6) ~= 0)
    totalHrs = totalHrs + 1;
end

t2 = t1-hours(12);

DownLoadError = 1;

while (DownLoadError && t1 >= t2)
% Store model data times
W = t1 + hours(0:3:totalHrs);

for j = 0:3:totalHrs

% The weather data URL
URL = 'http://nomads.ncep.noaa.gov/cgi-bin/filter.gfs.p00.pl?file=gfs.t';
URL = strcat(URL, num2str(hour(t1), '%02i'), '.pgrb2.1p00.f', num2str(j, '%03i'));
for k = 1:max(size(levels))
    URL = strcat(URL, 'klev_', num2str(levels(k), '%03i'), 'mb=on');
end
URL = strcat(URL, 'TMP=on&UGRD=on&VGRD=on&subregion=&leftlon=', num2str(leftlon), '&rightlon=', num2str(rightlon), '&toplat=', num2str(toplat), ...
    '&bottomlat=', num2str(bottomlat), '&dir=%2Fgfs.');
URL = strcat(URL, num2str(year(t1)));
URL = strcat(URL, num2str(month(t1), '%02i'));
URL = strcat(URL, num2str(day(t1), '%02i'));
URL = strcat(URL, num2str(hour(t1), '%02i'));
fileName = strcat(pwd(), '/deterministic_winds_', num2str(j), '.grib2');

% Download weather data
try
    outfilename = websave(fileName, URL);
    catch
        if (j==0)
            t1 = t1 - hours(6);
            totalHrs = totalHrs + 6;
            DownLoadError = 1;
            break;
        else
            return;
        end
    end
end

% Import weather data into MATLAB
nc = ncgeodataset(outfilename);

% Fill values for T, U and V
V(floor(j/3 + 1, :) = nc{'v-component_of_wind_isobaric'}(:)); %#ok<AGROW>
U(floor(j/3 + 1, :) = nc{'u-component_of_wind_isobaric'}(:)); %#ok<AGROW>
T(floor(j/3 + 1, :) = nc{'Temperature_isobaric'}(:)); %#ok<AGROW>
DownLoadError = 0;
end
end
end

if (DownLoadError == 1)
    return;
end

lat = nc{'lat'}(:);
lon = nc{'lon'}(:);
iso = nc{'isobaric'}(:)/100;
for j = 0:3:totalHrs
    outfilename = strcat(pwd(),'
deterministic_winds',',num2str(j),'.grib2');
    delete(outfilename);
    delete(strcat(outfilename,'.gbx9'));
    delete(strcat(outfilename,'.ncx'));
end
end
Appendix B. Matlab Code: Network Building

A Linear Interpolation in Time and Space

Code/InterpolateAllData.m

1 routes = {'KSUU-PHN'L', 'KTCM-CYQX', 'KTCM-KCHS', 'KRIV-KWRI'};
2
3 % lat/longs from airnav.com
4 latlongs_routes = [38.2645367, -121.9241315, 21.3178275, -157.9202627; %Travis lat/long, Honolulu
5 lat\].long
6 47.1376778, -122.4764750, 48.936944, -80.0405278; %McChord Lat/Long, Gander
7 lat/long (from skyvector)
8 33.8819433 -117.2590169, 40.0155833 , -74.5916991]; %March Air Reserve Air
9 base, JBMDL lat/long
10 [num_routes,˜] = size(latlongs_routes);
11 %mkdir Data
12 for i = 1:num_routes %create all folders based on route names before pulling data
13 FolderName = sprintf('%s ', routes{i}) %save in folder named after route
14 matfile = fullfile ('C:\\Users\smboo\\Desktop\Thesis (1)\\Thesis\\Data', FolderName);
15 ed(matfile)
16 addpath(genpath ('C:\\Users\smboo\\Desktop\Thesis (1)\\Thesis'))
17 files = dir ('*.mat');
18 for file = files
19 load(file.name);
20 [num_its,˜,˜,˜] = size(U)
21 tic
22 % issues with interpolation code, can only interpolate between two times, this is a workaround
23 % NOTE: this works for the short routes we are investigating during this
24 % research, will need to adjust for longer routes (i.e. routes > 6 hours)
25 if file.name(12) == 'D' %determine if its ensemble or deterministic data
26 model = 'D';
27 num_mem = 1;
28 for j = 3:num_its-2
29 [U_interp1, V_interp1] = time_alt_interp(U(j:j+1, :, :, :, :), V(j:j+1, :, :, :, :), model, num_mem);
30 [U_interp2, V_interp2] =
31                 time_alt_interp(U(j+1:j+2, :, :, :, :), V(j+1:j+2, :, :, :, :), model, num_mem);
32 U_interp = [U_interp1 ; U_interp2(2:end, :, :, :, :)];
33 V_interp = [V_interp1 ; V_interp2(2:end, :, :, :, :)];
34 end
35 else
36 model = 'E';
37 num_mem = 20;
38 [U_interp, V_interp] = time_alt_interp(U(2:end, :, :, :, :), V(2:end, :, :, :, :), model, num_mem);
39 end
40
```matlab
function [U_interp, V_interp] = time_alt_interp(U, V, model, num_mem)
if model == 'E' %if ensemble data
    x = [0:6]; %because time is in 6 hour timesteps, we have time 0 and time 6
    xi = [0:6]; % interpolating between 0 and 6 (included so that they are in the output vector)
    [num_times, num_alts, num_lats, num_long] = size(U);
    U_time = []; % initialize matrix for concatenating
    V_time = [];
    for m = 1 : num_times - 1
        for l = 1 : num_lats
            for k = 1 : num_alts
                for i = 1 : num_mem
                    y_U = U(m,m+1,i,j,k,l);
                    U_time_temp(:,i,j,k,l) = interp1(x,y_U,xi); % interpolate across U
                    y_V = V(m,m+1,i,j,k,l);
                    V_time_temp(:,i,j,k,l) = interp1(x,y_V,xi); % interpolate across V
                end
                if m > 1
                    [cur_row, U_time] = size(U_time); % ensure no duplicate rows
                    U_time(cur_row:cur_row+6:cur_row+6, j, k, l) = U_time_temp(:, j, k, l);
                    [cur_row, V_time] = size(V_time); % ensure no duplicate rows
                    V_time(cur_row:cur_row+6:cur_row+6, j, k, l) = V_time_temp(:, j, k, l);
                else
                    U_time(:, j, k, l) = U_time_temp(:, j, k, l); % add temporary to U with interpolation
                    V_time(:, j, k, l) = V_time_temp(:, j, k, l); % add temporary to V with interpolation
                end
            end
        end
    end
end
% altitude interpolation
[num_times, num_alts, num_lats, num_long] = size(U_time);
U_interp = zeros(num_times, 20, 11, num_lats, num_long);
V_interp = zeros(num_times, 20, 11, num_lats, num_long);
```

Code/time_alt_interp.m
for l = 1 : num_long
for k = 1 : num_lats

step_low_U = (U_time(:,:,2,k,l) - U_time(:,:,1,k,l))/5;
step_high_U = (U_time(:,:,3,k,l) - U_time(:,:,2,k,l))/5;
step_low_V = (V_time(:,:,2,k,l) - V_time(:,:,1,k,l))/5;
step_high_V = (V_time(:,:,3,k,l) - V_time(:,:,2,k,l))/5;

for incr = 0:10
    if incr <= 5
        U_interp(:,:,incr+1,k,l) = U_time(:,:,1,k,l) + incr*step_low_U; %Each new dim in num_alts will be 1000k increments from 25:35k
        V_interp(:,:,incr+1,k,l) = V_time(:,:,1,k,l) + incr*step_low_V; %Each new dim in num_alts will be 1000k increments from 25:35k
    else
        U_interp(:,:,incr+1,k,l) = U_time(:,:,1,k,l) + incr*step_high_U; %Each new dim in num_alts will be 1000k increments from 25:35k
        V_interp(:,:,incr+1,k,l) = V_time(:,:,1,k,l) + incr*step_high_V; %Each new dim in num_alts will be 1000k increments from 25:35k
    end
end
end
end
else
elseif model == 'D' %If deterministic data

x = [0;3]; %Because time is in 6 hour timesteps, we have time 0 and time 6
xi = [0:3]; %Interpolating between 0 and 6 (included so that they are in the output vector)
[num_times, num_alts, num_lats, num_long] = size(U);
U_time = []; %Initialize matrix for concatenating
V_time = []; %Initialize matrix for concatenating
for m = 1 : num_times-1
    for l = 1 : num_long
        for k = 1 : num_lats
            for j = 1 : num_alts
                y_U = U(m:m+1,i,j,k,1);
                U_time_temp(:,:,i,j,k,l) = interp1(x,y_U,xi); %Interpolate across U
            end
        end
        for j = 1 : num_alts
            y_V = V(m+1:m+1,i,j,k,1);
            V_time_temp(:,:,i,j,k,l) = interp1(x,y_V,xi); %Interpolate across V
        end
    end
end
if m > 1
    cur_row_U = size(U_time); %Ensure no duplicate rows
    U_time(cur_row_U:cur_row_U+3,:,:,k,l) = U_time_temp(:,:,j,k,l);
    cur_row_V = size(V_time);
    V_time(cur_row_V:cur_row_V+3,:,:,j,k,l) = V_time_temp(:,:,j,k,l);
else
    U_time(:,:,j,k,l) = U_time_temp(:,:,j,k,l); %Add temporary to U with interpolation
    V_time(:,:,j,k,l) = V_time_temp(:,:,j,k,l); %Add temporary to V with interpolation
end
end
end
end

%% altitude interpolation
[num_times, , num_alts, num_lats, num论述s] = size(U_time);
U_interp = zeros(num_times, 1, 11, num_lats, num论述s);
V_interp = zeros(num_times, 1, 11, num_lats, num论述s);
for l = 1:num论述s
  for k = 1:num_lats
    step_low_U = (U_time(:, :, 2, k, l) - U_time(:, :, 1, k, l)) / 2;
    step_high_U = (U_time(:, :, 3, k, l) - U_time(:, :, 2, k, l)) / 2;

    step_low_V = (V_time(:, :, 2, k, l) - V_time(:, :, 1, k, l)) / 2;
    step_high_V = (V_time(:, :, 3, k, l) - V_time(:, :, 2, k, l)) / 2;

    for incr = 0:10
      if incr < 2
        U_interp(:, :, incr+1, k, l) = U_time(:, :, 1, k, l) + incr * step_low_U; % each new dim in num论述s will be 1000k increments from 25:35k
        V_interp(:, :, incr+1, k, l) = V_time(:, :, 1, k, l) + incr * step_low_V; % each new dim in num论述s will be 1000k increments from 25:35k
      end
    end
  end
end
else
% % Truth Source
% time interpolation
x = [0; 6]; % because time is in 6 hour time steps, we have time 0 and time 6
xi = [0:6]; % interpolating between 0 and 6 (included so that they are in the output vector)
[num_times, , num_alts, num_lats, num论述s] = size(U);
U_time = []; % initialize matrix for concatenating
V_time = []; % initialize matrix for concatenating
for m = 1:num_times-1
  for l = 1:num论述s
    for k = 1:num_lats
      for j = 1:num_alts
        y_U = U(m:m+1, i, j, k, l);
        U_time_temp(:, i, j, k, l) = interp1(x, y_U, xi); % interpolate across U
        y_V = V(m:m+1, i, j, k, l);
        V_time_temp(:, i, j, k, l) = interp1(x, y_V, xi); % interpolate across V
      end
    end
  end
end
if m > 1
  [cur_row_U, , cur_row_U + 6, i, j, k, l] = size(U_time);
  U_time(cur_row_U + 1:i, j, k, l) = U_time_temp(:, i, j, k, l);
  [cur_row_V, , cur_row_V + 6, i, j, k, l] = size(V_time);
  V_time(cur_row_V + 1:i, j, k, l) = V_time_temp(:, i, j, k, l);
else
  U_time(:, i, j, k, l) = U_time_temp(:, i, j, k, l); % add temporary to U with interpolation
V_{time}(::,j,k,l) = V_{time,temp}(::,j,k,l); %add temporary to V with interpolation
end
end
end
end
time (::,::,j,k,l) = temp (::,::,j,k,l);

% add temporary to V with interpolation

end
end
end
end

end
end
end
end

% % altitude interpolation

[num_times, num_alts, num_lats, num_longs] = size(U_time);
U_interp = zeros(num_times, 1, 11, num_lats, num_longs);
V_interp = zeros(num_times, 1, 11, num_lats, num_longs);
for l = 1 : num_longs
for k = 1 : num_lats
step_low_U = (U_time(:,2,k,l) - U_time(:,1,k,l))/2;
step_high_U = (U_time(:,3,k,l) - U_time(:,2,k,l))/2;
step_low_V = (V_time(:,2,k,l) - V_time(:,1,k,l))/2;
step_high_V = (V_time(:,3,k,l) - V_time(:,2,k,l))/2;
for incr = 0:10
if incr <= 2
U_interp(:,incr+1,k,l) = U_time(:,1,k,l) + incr*step_low_U; % each new dim in num_alts will be 1000k increments from 25:35k
V_interp(:,incr+1,k,l) = V_time(:,1,k,l) + incr*step_low_V; % each new dim in num_alts will be 1000k increments from 25:35k
else
U_interp(:,incr+1,k,l) = U_time(:,1,k,l) + incr*step_high_U; % each new dim in num_alts will be 1000k increments from 25:35k
V_interp(:,incr+1,k,l) = V_time(:,1,k,l) + incr*step_high_V; % each new dim in num_alts will be 1000k increments from 25:35k
end
end
end
end
end

B Wind Calculations

Code/WindCalculs_26Oct.m

function [optimal_value, path, DG] = windcalcs(nlegs, mem_num, latlongmat, lat, U_interp, V_interp, lon, TAS, model, num_mem)

AC = 2; % use C-17 regression models
omega = 496.5; % AC gross weight estimate
PW = 5; % payload weight estimate
load RegressionCoeffs.mat % load Betas for Reiman regression models
temp = U_interp(:,mem_num,:,:,:,:);  \% look at one ensemble member at a time
U_mem = squeeze(U_temp);  \% reduce size of U since ensemble mem dimension went from 20 to 1 (now singular)

% Positive: West

mem = squeeze(U_temp);  \% reduce size of U since ensemble mem dimension went from 20 to 1 (now singular)

% Positive: South

[nhrs,~,~,~,~] = size(U_interp);  \% Determine U and V components along the route

% degree model, so need to look at each degree, so we can either:
% Used a weighted average based on the location of the actual data pt wrt to the upper/lower
latRnd = round(latlongmat(:,1));  \% just looking at the lower bnds for lat
longLw = floor(latlongmat(:,2));  \%
longUp = ceil(latlongmat(:,2));  \%

lon = mod(lon,360);  \% change lon coordinate from -180 to 180 to 0 to 360

for i = 1:nlegs+1  \% find idx of our route among all data
    longIdxLw(i) = find(lon == longLw(i));
    longIdxUp(i) = find(lon == longUp(i));
    latIdx(i) = find(lat == latRnd(i));
end

for i = 1:nlegs  \% determine U and V components along route using a weighted average between
    degrees; dims: time, altitude, leg
    U_route(:, :, i) = ((U_mem(:, :, latIdx(i+1)), longIdxLw(i+1))*(latlongmat(i+1,2) - longLw(i+1)) +
    U_mem(:, :, latIdx(i+1)), longIdxUp(i+1))*(1-(latlongmat(i+1,2) -
    longLw(i+1)))+((U_mem(:, :, latIdx(i)), longIdxLw(i))*(latlongmat(i,2) - longLw(i)) +
    U_mem(:, :, latIdx(i)), longIdxUp(i))*(1-(latlongmat(i,2) -
    longLw(i))))/2;
    V_route(:, :, i) = ((V_mem(:, :, latIdx(i+1)), longIdxLw(i+1))*(latlongmat(i+1,2) - longLw(i+1)) +
    V_mem(:, :, latIdx(i+1)), longIdxUp(i+1))*(1-(latlongmat(i+1,2) -
    longLw(i+1)))+((V_mem(:, :, latIdx(i)), longIdxLw(i))*(latlongmat(i,2) - longLw(i)) +
    V_mem(:, :, latIdx(i)), longIdxUp(i))*(1-(latlongmat(i,2) - longLw(i))))/2;
end

\% calculate headwind/tailwinds
WS = sqrt(U_route.^2 + V_route.^2);  \% Calculate windspeed m/s, converted to knots later
angle_W = atan2(-U_route, -V_route)*180/pi;  \% angle in degrees

for i = 2:nlegs+1  \% calculate the aircraft heading between waypoints i: from, j: to
    acHeading_orig(i-1,1) =
        mod(atan2d(sin(latlongmat(i-1,2)-latlongmat(i-1,1)),
            cos(latlongmat(i-1,1)))*cos(latlongmat(i-1,1))*sin(latlongmat(i-1,1)) -
            angle_W)*180/pi;  \% Wind Direction (degrees)
end

windfrom = atan2(U_route, -V_route)*180/pi;  \% Wind Direction (degrees)

for i = 1:nlegs  \% calculate new aircraft heading taking into account drift, the group speed, and the wind correction angle (pos to the right)
[ac_heading_corr (i, i), GS (i, i), windcorrangle (i, i)] = driftcorr(ac_heading_orig (i), TAS, windfrom (i, i), WS (i, i)); % corrected heading to stay on course, groundspeed (knots), the wind correcting angle in degrees

end

\[ \text{dist} = \sqrt{(\text{alt}_{\Delta} \cdot 2 + \text{GC}^2)}; \quad \text{assuming a straight line distance between points} \]

avg_time_leg = mean(mean(dist/TAS)); % avg time across all altitudes to fly 1 leg
flt_time = avg_time_leg;
for i = 2:nlags+1
    flt_time(i,1) = flt_time(i-1) + avg_time_leg; % sum the time at each leg
end

timestep_use = round(flt_time); % round to nearest integer and use that hour of data

% Create Network and Determine Optimal Path

for latlong = 1:nlags % Calculate the time to traverse each arc based on ground speed and the distance between them
    for row = 1:size(alt_delta,1)
        for col = 1:size(alt_delta,1)
            alpha_i = 24+row;
            alpha_j = 24+col;
            time_TAS = dist(row, col) / TAS; % time in hours to fly distance without winds
            time_GS = dist(row, col) / GS(timestep_use(row)+1, col, latlong); % time in hours to fly distance with winds
            dist_leg = dist(row, col) * time_GS / time_TAS; % ratio to determine equivalent distance of fuel used with winds in NM
            if row == col
                fuel{latlong}(row, col) = FuelCalc('cruise', AC, alpha_i, omega, PW, dist_leg);
            elseif row < col
                fuel_i = FuelCalc('cruise', AC, alpha_i, omega);
                fuel_j = FuelCalc('cruise', AC, alpha_j, omega);
                fuel{latlong}(row, col) = abs(fuel_j - fuel_i);
                dist_climb_i = Climb_reg_dist(1, AC) + Climb_reg_dist(2, AC) * alpha_i + Climb_reg_dist(3, AC) * alpha_i^2 + Climb_reg_dist(4, AC) * alpha_i^3 + Climb_reg_dist(5, AC) * omega + Climb_reg_dist(6, AC) * omega^2 + Climb_reg_dist(7, AC) * omega^3 + 10^(-6) * Climb_reg_dist(8, AC) * alpha_i^2 * omega^3
            end
        end
    end
end
\(10^\circ(-6)\cdot\text{Climb}_{\text{reg}}\cdot\text{dist}(9,\text{AC})\cdot\alpha_i\cdot\omega^3\); \(\text{determine distance to climb}\)
\[
dist_{\text{climb}j} = \text{Climb}_{\text{reg}}\cdot\text{dist}(1,\text{AC}) + \text{Climb}_{\text{reg}}\cdot\text{dist}(2,\text{AC})\cdot\alpha_j + \\
\text{Climb}_{\text{reg}}\cdot\text{dist}(3,\text{AC})\cdot\alpha_j + \\
\text{Climb}_{\text{reg}}\cdot\text{dist}(4,\text{AC})\cdot\alpha_j + \\
\text{Climb}_{\text{reg}}\cdot\text{dist}(5,\text{AC})\cdot\omega + \text{Climb}_{\text{reg}}\cdot\text{dist}(6,\text{AC})\cdot\omega^2 + \\
\text{Climb}_{\text{reg}}\cdot\text{dist}(7,\text{AC})\cdot\omega^3 + 10^\circ(-6)\cdot\text{Climb}_{\text{reg}}\cdot\text{dist}(8,\text{AC})\cdot\alpha_j - 2\cdot\omega^3 + \\
10^\circ(-6)\cdot\text{Climb}_{\text{reg}}\cdot\text{dist}(9,\text{AC})\cdot\alpha_j - 2\cdot\omega^3; \(\text{determine distance to climb}\)
\]
\[
dist_{\text{climb}} = dist_{\text{climb}j} - dist_{\text{climb}i};
\]
\[
\text{if dist}_{\text{climb}} < \text{dist}_{\text{leg}} \ % \text{if climb dist is < dist of leg, calculate fuel consumed on remaining dist as cruise}
\]
\[
\text{fuel}_{(\text{latlong})}(\text{row},\text{col}) = \text{fuel}_{(\text{latlong})}(\text{row},\text{col}) + \text{FuelCalc('cruis',\text{AC},
\alpha_i,\omega,\text{PW, dist}_{\text{leg}}-\text{dist}_{\text{climb}});
\]
\[
\text{elseif dist}_{\text{climb}} > \text{dist}_{\text{leg}} \ % \text{if climb dist is > dist of leg, only use a fraction of the total fuel for that climb}
\]
\[
\text{fuel}_{(\text{latlong})}(\text{row},\text{col}) = \text{fuel}_{(\text{latlong})}(\text{row},\text{col})\cdot\text{dist}_{\text{leg}}/dist_{\text{climb}};
\]
\[
\end\text{end}
\]
\[
\text{else}
\]
\[
\text{fuel}_i = \text{FuelCalc('descd',\text{AC}, \alpha_i, \omega);}
\]
\[
\text{fuel}_j = \text{FuelCalc('descd',\text{AC}, \alpha_j, \omega);}
\]
\[
\text{fuel}_{(\text{latlong})}(\text{row},\text{col}) = \text{abs}(\text{fuel}_j - \text{fuel}_i);
\]
\[
\text{dist}_{\text{descd}i} = \text{Descend}_{\text{reg}}\cdot\text{fuel}(1,\text{AC}) + \text{Descend}_{\text{reg}}\cdot\text{fuel}(2,\text{AC})\cdot\omega + \\
\text{Descend}_{\text{reg}}\cdot\text{fuel}(3,\text{AC})\cdot\omega^2 + \text{Descend}_{\text{reg}}\cdot\text{fuel}(4,\text{AC})\cdot\alpha_i + \\
\text{Descend}_{\text{reg}}\cdot\text{fuel}(5,\text{AC})\cdot\alpha_j\cdot\omega;
\]
\[
\text{dist}_{\text{descd}j} = \text{Descend}_{\text{reg}}\cdot\text{fuel}(1,\text{AC}) + \text{Descend}_{\text{reg}}\cdot\text{fuel}(2,\text{AC})\cdot\omega + \\
\text{Descend}_{\text{reg}}\cdot\text{fuel}(3,\text{AC})\cdot\omega^2 + \text{Descend}_{\text{reg}}\cdot\text{fuel}(4,\text{AC})\cdot\alpha_j + \\
\text{Descend}_{\text{reg}}\cdot\text{fuel}(5,\text{AC})\cdot\alpha_j\cdot\omega;
\]
\[
\text{dist}_{\text{descd}} = \text{dist}_{\text{descd}j} - \text{dist}_{\text{descd}i};
\]
\[
\text{if dist}_{\text{descd}} < \text{dist}_{\text{leg}} \ % \text{if descend dist is < dist of leg, calculate fuel consumed on remaining dist as cruise}
\]
\[
\text{fuel}_{(\text{latlong})}(\text{row},\text{col}) = \text{fuel}_{(\text{latlong})}(\text{row},\text{col}) + \text{FuelCalc('cruis',\text{AC},
\alpha_i,\omega,\text{PW, dist}_{\text{leg}}-\text{dist}_{\text{descd}});
\]
\[
\text{elseif dist}_{\text{descd}} > \text{dist}_{\text{leg}} \ % \text{if descend dist is > dist of leg, only use a fraction of the total fuel for that descend}
\]
\[
\text{fuel}_{(\text{latlong})}(\text{row},\text{col}) = \text{fuel}_{(\text{latlong})}(\text{row},\text{col})\cdot\text{dist}_{\text{leg}}/dist_{\text{descd}};
\]
\[
\end\text{end}
\]
\[
\text{end}
\]
\[
\text{ fuels } = []; \ % \text{initialize}
\]
\[
\text{start node } = []; \ % \text{initialize}
\]
\[
\text{end node } = []; \ % \text{initialize}
\]
\[
\text{for } i = 2:nlegs-1 \ % \text{create a vector for time to traverse nodes across cruise waypoints (omit nodes 1 and nlegs because we are forcing them to happen at the lowest altitude)}
\]
\[
\text{fuels } = \{\text{fuels fuel}([i,:])\};
\]
\[
\text{end}
\]
\[
\text{for } i = 2:nlegs-1 \ % \text{create a vector of "from" nodes that correspond to the times vector above}
\]
\[
\text{start node } = [\text{start node repmat([1+11*(i-1) 2+11*(i-1) 3+11*(i-1) 4+11*(i-1) 5+11*(i-1) 
\text{6+11*(i-1) 7+11*(i-1) 8+11*(i-1) 9+11*(i-1) 10+11*(i-1) 11+11*(i-1)])];
\]
\[
\text{end}
\]
\[
\text{start node } = [\text{start node (11+nlegs)-10:11+nlegs}];
\]
\[
\text{for } i = 23:11+nlegs+1 \ % \text{create "to" nodes that correspond to the "from" nodes and the times above}
\]
end_node = [end_node repmat(1,1,11)];

end

W = [fuel(1), fuels, fuel{nlegs}(:,1)]; % the arc weights for all nodes (correspond to the
times to traverse, for now)
DG = sparse([repmat(1,1,11), start_node],[12:end],W); %create the network
DG = [DG; zeros(size(DG,2)-size(DG,1),size(DG,2))]; % sparse arrays need the same number of rows
and cols (in order to implement graphshortestpath), so add an empty row to make the array
square

[optimal_value,path,pred] = graphshortestpath(DG,1,(11*nlegs)+1);
% optimal_value in kLbs

C Fuel Regression Equations

Code/FuelCalc.m

1 function fuel_consumed = FuelCalc(eq,AC, alpha, omega, PW, dist)
2 load RegressionCoeffs.mat %load Beta’s from Reiman regression models
3
4 if eq == 'climb'
5 % fuel to climb in Klbs
6 fuel_consumed = Climb_reg_fuel(1,AC) + Climb_reg_fuel(2,AC)*alpha + Climb_reg_fuel(3,AC)*alpha^2
7 + Climb_reg_fuel(4,AC)*alpha^3 + Climb_reg_fuel(5,AC)*omega + Climb_reg_fuel(6,AC)*omega^2 +
8 Climb_reg_fuel(7,AC)*omega^3 + 10^(-6)*Climb_reg_fuel(8,AC)*alpha^2*omega^3 +
9 10^(-6)*Climb_reg_fuel(9,AC)*alpha^2*omega^3;
10 elseif eq == 'descd'
11 % fuel to descend in Klbs
12 fuel_consumed = Descend_reg_fuel(1,AC) + Descend_reg_fuel(2,AC)*omega +
13 Descend_reg_fuel(3,AC)*omega^2 + Descend_reg_fuel(4,AC)*alpha +
14 Descend_reg_fuel(5,AC)*alpha*omega;
15 else
16 % fuel to cruise in Klbs
17 OW = PayloadAssumptions(1,AC); % operating weight
18 FRC = PayloadAssumptions(5,AC); % reserve/contingency fuel weight
19 FAH = PayloadAssumptions(6,AC) + PayloadAssumptions(7,AC); % alternate/holding fuel weight
20 A = SpecRange_reg(5,AC)/3;
21 B = (SpecRange_reg(4,AC)/2) + SpecRange_reg(5,AC)*(OW + FRC + FAH + PW) +
22 (SpecRange_reg(6,AC)/2)*alpha;
23 C = SpecRange_reg(1,AC) + SpecRange_reg(2,AC)*alpha + SpecRange_reg(3,AC)*alpha^2 +
24 SpecRange_reg(4,AC)*(OW+FRC+FAH+PW)+SpecRange_reg(5,AC)*(OW+FRC+FAH+PW)^2 +
25 SpecRange_reg(6,AC)*alpha*(OW+FRC+FAH+PW);
26 D = -dist;
27 %fuel_consumed = -B/(3*A) -
28 1/(3*A)*((1/2)*((2*B*3-9*A+B+C+27*A^2*D+sqrt((2*B*3-9*A+B+C+27*A^2*D)^2-4*(B^2-3*A+C)^3))^(1/3))
29 - 1/(3*A)*((1/2)*((2*B*3-9*A+B+C+27*A^2*D+sqrt((2*B*3-9*A+B+C+27*A^2*D)^2-4*(B^2-3*A+C)^3))^(1/3))

59
commonterm1 = 2*B^3 - 9*A*B*C + 27*A^2*D;
commonterm2 = 4*(B^2 - 3*A*C)^3;
cubterm1 = ((1/2)*(commonterm1 + sqrt(commonterm1^2 - commonterm2)))^(1/3);
cubterm2 = ((1/2)*abs(commonterm1 - sqrt(commonterm1^2 - commonterm2)))^(1/3);
cubterm2 = sign(commonterm1 - sqrt(commonterm1^2 - commonterm2))*cubterm2; %/c matlab yields a complex number
fuel_consumed = -B/(3*A) - (1/(3*A))*cubterm1 - (1/(3*A))*cubterm2;
end
end
Appendix C. Matlab Code: The Models

A Deterministic Model

Code/SolveRoutes_Det.m

```matlab
function [optimal_value, path] = SolveRoutes_Det(latlongs_routes, route_num, srt_trial, stepsize, num_trial, TAS, U_interp, V_interp, lat, lon)

model = 'D'; % using deterministic data

sol_mat = zeros(num_trial, 3); % ensemble number | nleg | optimal value
nleg = srt_trial:stepsize:num_trial+1; % ensemble number
num_mem = 1;

for i = 1:num_trial
    path_temp = zeros(20, nleg(i)+1);
    sol_mat_temp = [];

    latlongmat = gcwaypts(LatA, LongA, LatB, LongB, nleg(i));
    latlongmat(:,2) = mod(latlongmat(:,2),360);

    [optimal_value, path, DG] = WindCalcs_26Oct(nleg(i), 1, latlongmat, lat, U_interp, V_interp, lon, TAS, model, num_mem);
    sol_mat_temp(1,1) = 1;
    sol_mat_temp(1,2) = nleg(i);
    sol_mat_temp(1,3) = optimal_value;
    path_temp(1,:) = path;
    path_mat(:,1:nleg(i)+1) = path_temp;
    sol_mat = [sol_mat; sol_mat_temp];
end
end
```

B Homan Model

Code/SolveRoutes_Mean.m

```matlab
function [optimal_value, path] = SolveRoutes_Mean(latlongs_routes, route_num, srt_trial, stepsize, num_trial, TAS, U_interp, V_interp, lat, lon)

model = 'E'; % using ensemble data

num_mem = 20;

% based on which route analyzing
```

61
function [path_mat, optimal_vals, unique_routes, num_routes, cost] = SolveRoutes_IID(latlongs_routes, route_num, strt_trial, stepsize, num_trial, TAS, U_interp, V_interp, lat, lon)

model = 'E'; % using ensemble data
num_mem = 20;

% based on which route analyzing
LatA = latlongs_routes(route_num,1);
LongA = latlongs_routes(route_num,2);
LatB = latlongs_routes(route_num,3);
LongB = latlongs_routes(route_num,4);

sol_mat = zeros(num_trial, 3); % ensemble number | n legs | optimal value
nlegs = strt_trial:stepsize:num_trial+strt_trial;
sol_mat = [];
for i = 1:num_trial
    path_temp = zeros(20, nlegs(i) +1);
sol_mat_temp = [];
    for j = 1:num_mem
        latlongmat = gcwaypts(LatA, LongA, LatB, LongB, nlegs(i));
        latlongmat(:, 2) = mod(latlongmat(:, 2), 360);
        mean_U = mean(U_interp, 2);
        mean_V = mean(V_interp, 2);
        [optimal_value, path, DG] = WindCalcs_26Oct(nlegs(i), 1, latlongmat, lat, mean_U, mean_V, lon, TAS, model, num_mem);
        sol_mat_temp(1, 1) = 1;
        sol_mat_temp(1, 2) = nlegs(i);
        sol_mat_temp(1, 3) = optimal_value;
    end
end
[optimal_value, path, DG] = WindCalcs_26Oct(nlegs(i), mem(j), latlongmat, lat, U_interp, V_interp, lon, TAS);
sol_mat_temp(j, 1) = j;
sol_mat_temp(j, 2) = nlegs(i);
sol_mat_temp(j, 3) = optimal_value;
path_temp(j,:) = path;
end
path_mat(:,1:nlegs(i)+1,i) = path_temp;
sol_mat = [sol_mat; sol_mat_temp];
end
optimal_vals = sol_mat(:,3);

%%% ncol = 0;
all_unique_routes = [];
for l = 1:num_trial
    unique_routes = unique(path_mat(:,1,:),'rows');  \% determine which paths are unique across ensemble members for each val of nlegs
    if num_trial > 1
        unique_routes(:,l*stepsize+2:end) = [];  \% remove 0's from nlegs being different
    end
    [numRoutes,~] = size(unique_routes);  \% determine the number of unique paths
    latlongmat = gcwaypts(LatA,LongA,LatB,LongB,nlegs(l));
    latlongmat(:,2) = mod(latlongmat(:,2),360);
    for k = 1:num_routes
        for i = 1:20
            cost_vect = [];
            [~,~,DG] = WindCalcs_26Oct(nlegs(i), mem(i), latlongmat, lat, U_interp, V_interp, lon, TAS, num_mem);
            for j = 2:nlegs(l)+1
                cost_vect = [cost_vect; full(DG(unique_routes(k,j-1),unique_routes(k,j)))];
            end
            cost((20*(k-1)+i),ncol+1:ncol+2) = [sum(cost_vect) k];
        end
    end
    num_routes_vec(l) = num_routes;
end
[~,ncol] = size(cost);
Bibliography


3. AMC/A3W. WX White Paper - Draft 1. amc.a3w@scott.af.mil, 2005.


31. MathWorks. MATLAB version 8.5.0.197613 (R2016a), 2016.


**REPORT DOCUMENTATION PAGE**

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1244, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>(DD-MM-YYYY)</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>22-03-2018</td>
<td>Master’s Thesis</td>
<td>Sept 2017 — Mar 2018</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHORTEST PATH ACROSS STOCHASTIC NETWORK WITH CORRELATED RANDOM ARCS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5a. CONTRACT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5b. GRANT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5d. PROJECT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5e. TASK NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stephanie M.Boone, 1st Lt, U.S. Air Force</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Force Institute of Technology</td>
</tr>
<tr>
<td>Graduate School of Engineering and Management</td>
</tr>
<tr>
<td>(AFIT/EN)</td>
</tr>
<tr>
<td>2950 Hobson Way</td>
</tr>
<tr>
<td>WPAFB OH 45433-7765</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFIT-ENC-MS-18-M-109</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMC Weather Operations and Plans Branch</td>
</tr>
<tr>
<td>402 Scott Drive, Unit 3A1</td>
</tr>
<tr>
<td>Scott AFB, IL 62225-5302</td>
</tr>
<tr>
<td>COMM 618-229-3636</td>
</tr>
<tr>
<td>Email: <a href="mailto:frederick.wirsing@us.af.mil">frederick.wirsing@us.af.mil</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMC/A3AW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION / AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTRIBUTION STATEMENT A:</td>
</tr>
<tr>
<td>APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper introduces a new approach to identify the shortest path across a stochastic network with correlated random arcs utilizing nonparametric samples of arc lengths. This approach is applied to find optimal aircraft routes that minimize expected fuel consumption for a given airspeed utilizing predicted wind output from NWP ensemble models. Results from this new methodology are then compared to the current fuel minimization route planning method that utilizes deterministic NWP wind data for arc lengths. Comparisons are also made to other previously proposed alternative fuel minimization methodologies that utilize mean and median wind data calculated from NWP ensemble wind data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT U</td>
</tr>
<tr>
<td>b. ABSTRACT U</td>
</tr>
<tr>
<td>c. THIS PAGE U</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lt Col A. Geyer, AFIT/ENC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19b. TELEPHONE NUMBER (include area code)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(937) 255-6565, x4584; <a href="mailto:Andrew.Geyer@afit.edu">Andrew.Geyer@afit.edu</a></td>
</tr>
</tbody>
</table>

*Standard Form 298 (Rev. 8–98)*

Prescribed by ANSI Std. Z39.18