Facilitating User Understanding of Optimizations: A Case Study of Channel Route Network Planning

by

Deb Prasad Dasgupta
S.B., Massachusetts Institute of Technology (2002)
Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Computer Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2003

Author ....
Department of Electrical Engineering and Computer Science
May 21, 2003

Certified by ....
Dr. Randall Davis
Professor of Electrical Engineering and Computer Science
MIT Artificial Intelligence Laboratory
Thesis Advisor

Certified by ....
Dr. Michael E. Cleary
Principal Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Facilitating User Understanding of Optimizations: A Case Study of Channel Route Network Planning

by

Deb Prasad Dasgupta

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

As more complex and important tasks become automated, the human-system interface is becoming more prominent. It is necessary for users to trust the systems performing these tasks; otherwise, they are unlikely to use the system. Automated planning is one such task. An integral part of planning systems using an Operations Research approach is the use of optimization techniques to create plans. In problems of realistic size, the solution process of the optimization is too complex to follow in detail, so it is not possible for the user to evaluate the effectiveness of the solution. For instance, the Channel Route Network Planning System uses an optimization to create plans for the shipment of cargo between military bases around the world. Although it chooses the optimal plan for a given set of inputs, its users could plan more effectively if they better understood the underlying decision space of its optimization and had ready access to the details of the plans it generates.

This thesis presents ChRIS, the Channel Route Information System. ChRIS is designed to enable users to gain insight into plans developed by results of the Channel Route Network Planning System. It helps users understand the internal structure of the individual plans and illustrates the differences between multiple plans, thereby helping users to understand the optimization, which we believe will engender trust in the system's choice of optimal plans.

Thesis Advisor: Dr. Randall Davis
Title: Professor of Electrical Engineering and Computer Science, MIT Artificial Intelligence Laboratory

Technical Supervisor: Dr. Michael E. Cleary
Title: Principal Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.

Acknowledgments

First I would like to thank Dr. Michael E. Cleary for making this opportunity possible. Throughout my summers as a co-op student and this year as a Draper Fellow, he has provided me with invaluable technical advice and guidance at any time that I needed it. His encouragement to pursue the problems I found interesting in a new and creative way made it easy for me to immerse myself in my work and get the most out of this experience.

I would also like to thank Professor Randy Davis. I appreciate the time and effort he put into supervising my thesis and the enthusiasm he brought to our discussions. He always made sure I understood the big picture surrounding what I was doing, and he was the source of many of the ideas that I used to make my thesis better.

I must also thank the Charles Stark Draper Laboratory for its commitment to education and research expressed though the Draper Fellow program. All of the people at Draper, especially Chris Nielsen and Dr. Stephan Kolitz, have always been helpful whenever I needed anything.

Next I have to thank my family. Without them, none of this would have been possible. Their unwavering love and support make all the late nights and hard work entirely worth it. Everything that I have done to this point, and everything from here on, is for them.

Finally, there are my friends. Richa has always been there to lean on, always knowing what I am thinking, especially when I'm too tired to say it. My Toons, both new and old, have always made sure that I don't work too hard and that I enjoy the things we all love. My friends are all amazing people, and they have given me memories that I will cherish for a lifetime.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Research and Development Contract IRD-03-1-5036, Variable Initiative Control.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

This thesis is dedicated to my father, whose memory inspires me in everything that I do.

Deb Dasgupta

Contents

1 Introduction 17

1.1 Motivation . . . 17

1.2 Concrete Example . . . 20

1.3 Structure of Thesis . . . 21

2 Background and Previous Work 23

2.1 Optimization and Shadow Prices . . . 23

2.2 Visualization Techniques . . . 27

2.3 HSI/GUI Design . . . 29

2.4 Visualization of Optimization Results . . . 31

3 Current Channel Route Network Planning Process 33

3.1 Introduction . . . 33

3.2 Problem Domain . . . 34

3.2.1 Airports . . . 34

3.2.2 Planes . . . 34

3.2.3 Cargo . . . 35

3.2.4 Flight Crews . . . 36

3.2.5 Plan Structure . . . 36

3.3 Current Manual Process . . . 37

3.4 Automated Optimization . . . 38

4 Design of the System 43

4.1 Introduction 43
4.2 Preparation for System Use 44
4.3 Comparing Multiple Plans 45
    4.3.1 Five-Dimensional Graph 46
    4.3.2 Metric Selection Tool 50
    4.3.3 Legend of Five-Dimensional Graph 51
        4.3.3.1 Non-Cartesian Axis Scales 53
        4.3.3.2 Zooming and Resetting Five-Dimensional Graph . . 55
        4.3.3.3 Average Parameter Values and Metric Scores 58
    4.3.4 Pop up Menu 59
4.4 Viewing a Single Plan 62
    4.4.1 Options Selector 63
    4.4.2 Mission Start Times 66
    4.4.3 Individual Missions 67
    4.4.4 Score Differentials 71
        4.4.4.1 One-Dimensional Score Differentials 72
        4.4.4.2 Two-Dimensional Score Differentials 78
    4.4.5 Available Planes 83
    4.4.6 Total Cargo and Final Cargo Locations 84
    4.4.7 Plan Scores 87
    4.4.8 Flying Hours 88
    4.4.9 Help 88

5 Example of System Use 91

6 Implementation 119

6.1 Introduction 119
6.2 Package single 121
6.3 Package main 124
6.4 Package util 127
    6.4.1 Processing of CRNPS Output Files by Plan Class 129

6.4.2 Processing of Output Files by Starts Class 130
6.5 Compiling and Running ChRIS 133

7 Conclusion 135
7.1 Summary and Contributions 135
7.2 Future Work 136

A Glossary 141

List of Figures

1-1 Costs of flying between pairs of airports 20

2-1 Explanation of Shadow Prices 28

3-1 The current channel route planning process (taken from [9]) 38
3-2 Sample mission from *missions_in_solution.out* 41
3-3 File specifying possible values for optimization inputs 42

4-1 Example file specifying probability of resource availability 45
4-2 Plan comparison tool set 47
4-3 *Five-Dimensional Graph* display 49
4-4 *Metric Selection Tool* 51
4-5 Multiple *Five-Dimensional Graphs* visible at once 52
4-6 *Legend* of the *Five-Dimensional Graph* 53
4-7 *Five-Dimensional Graph* with rectangular region selected 56
4-8 *Five-Dimensional Graph* zoomed in on selected region of Figure 4-7 . 57
4-9 Display of average parameter and metric values for selected region of Figure 4-7 59
4-10 Two average parameter and metric value displays 60
4-11 *Pop Up Menu* showing identifying information 61
4-12 *Pop Up Menu* offering plan details 61
4-13 Single plan visualization tool set 64
4-14 *Options Selector* 65
4-15 *Mission Start Times* tool 68

4-16 Visualization of missions leaving CHS on day 24 using C17 planes . . 69
4-17 Legend for *Individual Missions* display 70
4-18 Detail of Figure 4-16 showing planes adding significant amounts of cargo at NGU 71
4-19 Detail of Figure 4-16 illustrating that the displayed lines for four flights are overlapped from CHS to DOV and then divide from there to separate destinations. 71
4-20 Example of Score Differentials 73
4-21 Use of *One-Dimensional Score Differentials* Selection Tool 74
4-22 *One-Dimensional Score Differentials* 75
4-23 *Two-Dimensional Score Differentials* Selection Tool 79
4-24 *Two-Dimensional Score Differentials* 81
4-25 *Available Planes* tool showing available and used planes 83
4-26 *Total Cargo* display showing total cargo shipped by one plan 85
4-27 *Final Cargo Locations* display showing final planned cargo locations . 86
4-28 *Plan Scores* report for a plan 87
4-29 *Flying Hours* report showing flying hours provided by the plan 89
4-30 *Help* tool 90

5-1 Flowchart of example interaction 92
5-2 Initial *Metric Selection Tool* viewed by user 93
5-3 Initial *Five-Dimensional Graph* viewed by user 94
5-4 Second *Five-Dimensional Graph* viewed by user 95
5-5 Selection of upper left cluster 96
5-6 Average information for two clusters 97
5-7 View of top left corner of Figure 5-4 when zoomed in too far 99
5-8 View of top left corner of Figure 5-4 when not zoomed in enough . . . 100
5-9 Correctly zoomed in view of top left corner of Figure 5-4 101
5-10 First plan explored by user 102
5-11 Identifying information *Pop Up Menu* for first plan explored by user . 103

5-12 Selection of *Options Selector* from options *Pop Up Menu* for first plan explored by user 104
5-13 *Options Selector* for first plan explored by user 105
5-14 *Mission Start Times* for first plan explored by user 106
5-15 Display of *Individual Missions* leaving CHS on day 2 for first plan explored by user 107
5-16 *Final Cargo Locations* report for first plan explored by user 108
5-17 *Total Cargo* display for first plan explored by user 108
5-18 *Help* tool viewed by user 109
5-19 *Flying Hours* display for first plan explored by user 109
5-20 *Plan Scores* display for first plan explored by user 110
5-21 *Available Planes* display for first plan explored by user 110
5-22 Second plan explored by user 111
5-23 Third plan considered by user 113
5-24 *One-Dimensional Score Differential* graphs viewed by user 114
5-25 *Two-Dimensional Score Differential* graphs viewed by user 116
5-26 Close up view of *Two-Dimensional Score Differential* graphs 117

6-1 UML Class Diagram for entire implementation 120
6-2 Structure of *wrap_arcs.out* 129
6-3 Sample file listing flight legs for missions leaving CHS on day 10 . . . 132

List of Tables

3.1 Characteristics of planes flying channel route missions 35
3.2 Potential objective functions to be minimized 40
3.3 Outputs of the optimization 40

4.1 Descriptions of the seven metrics used to score a plan 46
4.2 Listing of information accessible from *Pop Up Menu* 62
4.3 Listing of information accessible from the *Options Selector* 65

6.1 Classes in package single 124
6.2 Classes in package main 127
6.3 Classes in package util 129

Chapter 1

Introduction

The system presented in this thesis provides tools that allow users of planning systems to gain insight into the results of what might otherwise be black-box optimizations. The system's interface presents the nature of the selected solution and allows the user to explore the ramifications of switching to a new solution, which is often necessary in the real world. This information is integral to building understanding of the solution and confidence in the plan to motivate its execution. This chapter motivates this need for understanding by presenting a concrete example of a planning problem solved with optimization techniques.

1.1 Motivation

An integral part of planning systems using a Operations Research approach is the use of optimization over the resources available, constraints and demands of the problem, and the desired goals [5]. The optimization routine determines the most effective allocation of resources, which is then used as the plan. Hereafter, this will be referred to simply as an optimization. For example, the Joint Forces Air Component Commander (JFACC) system, developed at the Charles Stark Draper Laboratory, plans missions by using an optimization to assign military resources to targets throughout a dynamic battle situation [3]. Another example is the planner for the UPS Next-Day Air delivery network, which uses an optimization to determine how packages should be routed to their destinations [2].

Even though an optimization selects the best plan for the situation, it is frequently very hard for users who are unfamiliar with optimization techniques to understand that solution and how it compares to other possible solutions. Although the users may understand the problem domain, they likely do not know or understand how the problem was formulated (eg. resources, constraints, and goals) and are unlikely to be familiar with the solution technique used. As a result, it may be difficult for them to understand the plan generated, even though it may be the best course of action.

There are many issues regarding human understanding of planner results. This work considers two of them: understanding of the structure of the plan, and explaining the ramifications of changing plans. The structure of the plan refers to what will actually occur if the plan is executed. This is especially important for planners such as the JFACC system, which is an example of systems which, when fielded, would place life at risk and may use many expensive resources. A commanding officer will be reluctant to order his men to execute a plan in a dangerous environment, potentially spending a lot of resources, when he does not understand whether and how the plan will do what he needs it to do. Giving users the ability to understand the recommendation that the optimization is making will help them achieve the necessary level of comfort and confidence in the plan. While most easily seen in a system such as the JFACC system, this is true of all optimizations. A user will be more willing to accept the optimization's recommendations if he understands whether and how it achieves his goals.

The second issue explored is the collective ramifications of a change in plan. In general, optimizations take a set of inputs that describe the current situation and produce a solution based on those inputs. This solution will work as long as the current situation does not change. However, in the real world the situation frequently will change, potentially rendering the current solution less than optimal in the new situation, or perhaps even impossible to execute, requiring the user to switch to a different solution. Switching solutions could be very expensive and time consuming. As the time for plan execution nears, there is less time available for replanning. It would be beneficial to the user to be able to see how much one plan compares to a different solution corresponding to a different set of inputs. For example, based upon this information, the user could choose to execute a solution with a slightly higher cost because there is another plan of similar cost that could also be used, which eliminates the need to replan in the face of changes to the set of inputs.

This thesis explores both of these issues, understanding the optimization solution and indicating the cost of switching to a new solution, in a case study of the Channel Route Network Planning System (CRNPS), an optimization-based planning system. This system plans the shipment of cargo around what is called the Channel Route Network (CRN). The CRN is the primary means of shipping supplies to many military bases around the world. This research project determined what information about CRN plans to show to the user in order to achieve understanding and to demonstrate the effects of change. Having identified this information, novel ways of displaying the information to the user were created. In this work, we are more interested in creating interfaces that show the most important information about the optimization, and are less concerned with issues of exactly how that information is presented to the user. Although we believe usability is important to interface design, the primary issue here was determining what information is required to understand optimization solutions and outputs. We believe that with the understanding gained by using the interface, prospective users will feel confident that the plans produced by the optimization will achieve their goals.

As more complex and important tasks become automated, the human-system interface is becoming more prominent. It is necessary for users to trust the systems performing these tasks; otherwise, they are unlikely to use the system. We believe that understanding the results of the optimization will enable the user to trust the output of the system. This study of a single system gives some tools for providing general insight into the information users need in order to understand and consequently trust optimization-based planning systems.

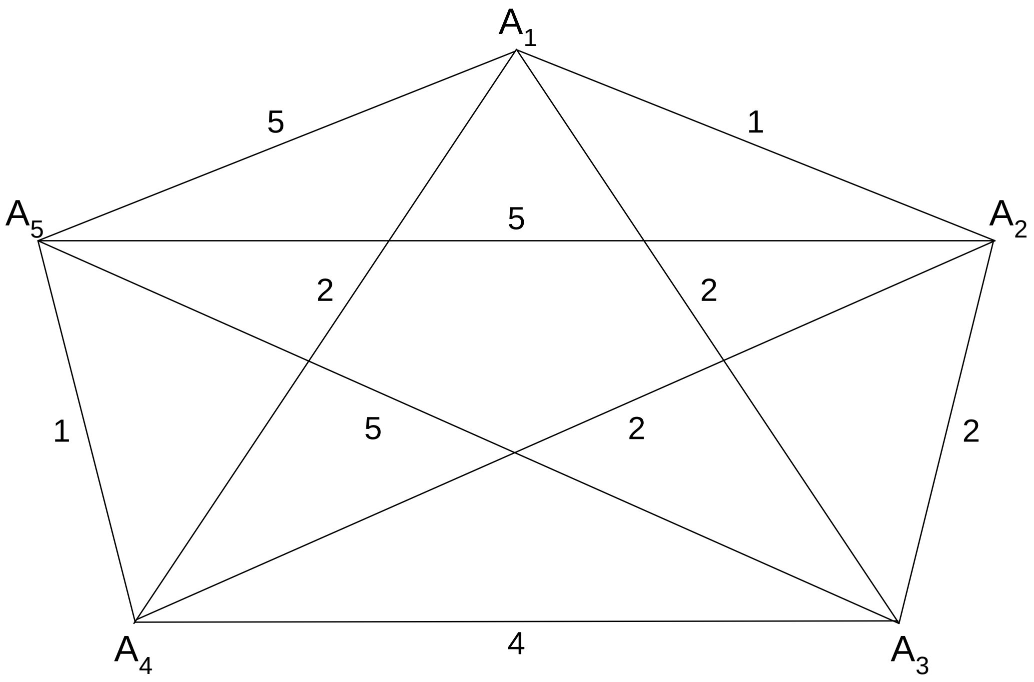


Figure 1-1: Costs of flying between pairs of airports

1.2 Concrete Example

Consider the following example: there are five airports, A_1, A_2, A_3, A_4, and A_5, and four commodities, numbered C_1, C_2, C_3, and C_4, to be shipped around this network of airports. All commodities begin at A_1. Commodity C_1 is destined for airport A_2, C_2 for A_3, C_3 for A_4, and C_4 for A_5. There are three planes available to carry the cargo, P_1, P_2, and P_3, and they are all based at A_1. The total weight of the four commodities is less than the carrying capacity of each of the planes. Flying between airports has a cost that is shown on the arcs in Figure 1-1. The task of an optimizing planner is to determine how to ship the commodities to their destinations with minimal cost.

The solution to this simple problem can also be found by inspection. The optimal solution is to have plane P_1 fly to airport A_2 carrying commodity C_1, P_2 fly to A_3 carrying C_2, and P_3 fly to A_4, and then to A_5, carrying C_3 and C_4. The total cost of this plan is 6. Since a hypothetical user could come up with the solution himself, he could be sure that the optimization's solution achieves his goal of getting all of the cargo to its destination with minimal cost. There are two things that make solution and understanding by inspection possible. First, the size of the problem is very small, and second, all of the information specifying the problem and its solution is accessible to the user. He can use this information to follow the calculations of the optimization.

Next, assume that plane P_2 is very old and could be lost at any time due to failure. In the event that it does fail, another plane would be needed to fly commodity C_2 to airport A_3. The user can easily come up with another plan using just P_1 and P_3. In this plan, P_1 flies to A_2, and then to A_3, carrying C_1 and C_2. P_3 still files to A_4, and then to A_5, carrying C_3 and C_4. Notice that the cost of this plan is still 6. Knowing that P_2 is likely to fail, the user could choose this second plan originally and not worry about losing P_2 and also not lose any money. Again, the small and simple nature of this problem allow the user to consider issues such as the cost of losing a plane.

However, real world problems are never this simple. A realistic problem of this sort may involve 30 airports, 1000 cargo commodities, and 60 planes. In this case, following all of the calculations is very difficult. For example, the number of inter-airport lines in Figure 1-1 would grow from 10 to almost 450, and there is a large growth in the number of other constraints. This more realistic problem is far too big to be solved by hand; an automated optimization must be used. In this case, the user is isolated from the solution process, which becomes more opaque and harder to understand, and to trust. The system and user interface presented in this thesis provides the transparency and explanation of the solution and its defining properties that provides a grounds for trust.

1.3 Structure of Thesis

This thesis presents ChRIS, the Channel Route Information System. ChRIS is a system designed to help users understand the results of the CRNPS. It does this by presenting information about the optimization solution in an informative yet easy to understand format.

Chapter 2 presents the background and previous work relevant to the design and implementation of ChRIS. It includes information about the use of optimization for planning, and about design principles for human-system interfaces and graphical user interfaces. Chapter 3 describes the current channel route network planning process. It describes the physical nature of the network, the manual planning process that is currently used, and the optimization that was designed to automate part of the planning process.

Chapter 4 presents the design of ChRIS. It describes the layout, function, and means of access of each element of the system. Chapter 5 is a user manual for ChRIS that explains how to use each element of the system. It motivates the design of each element and the information that is presented.

Chapter 6 describes the Java implementation of ChRIS. For each package in the implementation, all of the classes are briefly described. There is also a guide to compiling and running ChRIS. Chapter 7 concludes the thesis by summarizing the work done, identifying the contributions of the work, and outlining some areas for future work.

Chapter 2

Background and Previous Work

This chapter presents the background information relevant to the work done for this thesis. First, general optimization theory and shadow prices are discussed. For further details, see Tsitsiklis [5]. Then methods of data visualization are presented. Third, guidelines in developing human computer interfaces are discussed. Finally, there is a discussion of the literature search for visualization for optimization.

2.1 Optimization and Shadow Prices

In a linear optimization, the goal is to optimize an *objective function* [5]. The variables over which the objective function is optimized are called the *decision variables*. The decision variables are subject to some *constraints* that limit the values they can take in the optimal solution. Each of the decision variables has an associated *cost*, and the goal is to minimize the total cost (the objective function). In other words, optimization techniques search for an assignment of values to the decision variables that minimizes the total cost while obeying the constraints. The mathematical details of how this is done are presented below.

The objective function and constraints are all linear functions of the decision variables. The constraints take the form

$$\mathbf{a}'\mathbf{x} = b \tag{2.1}$$

where $\mathbf{a}$ is an $n \times 1$ vector of coefficients, $\mathbf{x}$ is an $n \times 1$ vector of decision variables, b is a scalar input, and $\mathbf{a}^{'}$ is the transpose of $\mathbf{a}$. For example, let $n = 2$, $\mathbf{a} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, and $b = 4$. The constraint in Equation 2.1 has the expanded form

$$3x_1 + 2x_2 = 4 \tag{2.2}$$

which is a linear function of the decision variables.

There are m constraints on the n decision variables, each of which limits the possible assignments to the decision variables. The constraints are often arranged in an $m \times n$ matrix and an $m \times 1$ vector of inputs. Let $\mathbf{A}$ be the $m \times n$ matrix of constraints, $\mathbf{b}$ be an $m \times 1$ input vector, $\mathbf{x}$ be an $n \times 1$ vector of decision variables, and $\mathbf{c}$ be an $n \times 1$ cost vector. Each row of $\mathbf{A}$ contains the transpose of one of the coefficient vectors ($\mathbf{a}$), and each element of $\mathbf{b}$ is one of the scalar inputs (b). An optimization is formulated to find the minimum cost linear combination of the columns of $\mathbf{A}$ that synthesizes the input, i.e.

$$\begin{aligned} &\textit{minimize} && \mathbf{c}^{'}\mathbf{x} && (2.3) \\ &\textit{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} && (2.4) \\ & && \mathbf{x} \geq \mathbf{0} && (2.5) \end{aligned}$$

In other words, the optimization finds the minimum cost assignment to the decision variables that is allowed by the constraints.

A *basis* of a space is the set of linearly independent vectors that span it. In optimizations, the space is the column space of the matrix $\mathbf{A}$, or the space spanned by the columns of $\mathbf{A}$. Let $\mathbf{B}$ be the $m \times m$ matrix whose columns are those columns of $\mathbf{A}$ that form a basis for the column space of $\mathbf{A}$. Also let $\mathbf{x_B}$ be the set of entries from $\mathbf{x}$ that correspond to the basic columns of $\mathbf{A}$, and let c_B be the corresponding entries from $\mathbf{c}$. The values of the elements of $\mathbf{x_B}$ are defined as follows:

$$\mathbf{x_B} = \mathbf{B^{-1}b} \tag{2.6}$$

If a basis satisfies the following two conditions with a given $\mathbf{b}$ and $\mathbf{c}$, then it is optimal:

$$\mathbf{x_B} \geq \mathbf{0} \tag{2.7}$$

$$\mathbf{c}' - \mathbf{c_B'}\mathbf{B^{-1}A} \geq \mathbf{0} \tag{2.8}$$

Note that the elements of $\mathbf{x}$ not in $\mathbf{x_B}$ will always equal zero, because the nonbasic columns of $\mathbf{A}$ are not needed in the optimal linear combination of the columns of $\mathbf{A}$ that synthesizes $\mathbf{b}$. In this case, the value of the objective function is:

$$\begin{aligned} cost &= \mathbf{c_B'x_B} && (2.9) \\ &= \mathbf{c_B'B^{-1}b} && (2.10) \end{aligned}$$

The cost function is a piecewise linear function of the input vector $\mathbf{b}$, and there are discontinuities between the pieces. Recall that $\mathbf{B}$ is comprised of the columns of $\mathbf{A}$ that are needed to synthesize $\mathbf{b}$. Changes to $\mathbf{b}$ may necessitate changes to $\mathbf{B}$. In other words, a different set of the columns of $\mathbf{A}$ may be necessary to synthesize the changed $\mathbf{b}$. A change to $\mathbf{B}$ is called a change of basis. The values of $\mathbf{b}$ at which there is a change of basis are the locations of the discontinuities in the cost function.

Shadow prices are defined as the amount by which the objective function changes for a unit change in an input. There is one shadow price for each input to the optimization (i.e. for each element of $\mathbf{b}$). Shadow prices indicate the sensitivity of objective value to changes in the inputs to the optimization. A high shadow price means that small changes to that input will cause large changes to the objective value. An interpretation of a shadow price is the partial derivative of the objective function with respect to a single input. In other words, shadow prices describe how the objective function changes with changes in the inputs. The partial derivative is not defined at a discontinuity, and it may be different on either side of the discontinuity. Therefore, if a change of basis exists between two points, where a point is a value of **b**, there is a discontinuity in the cost function between them, and no claims can be made about the difference in objective value over the entire interval between them being related to one shadow price. However, if there is no change of basis between the two values of **b**, there is no discontinuity in the cost function between them, and the shadow price or partial derivative relates how the objective value changes over the entire interval between them.

For a given basis, the vector of shadow prices **p** is given by:

$$\mathbf{p} = \mathbf{c}_{\mathbf{B}}^{'}\mathbf{B}^{-\mathbf{1}} \tag{2.11}$$

There is one set of shadow prices per pair of basis and cost vector **c**. When changes are made to the input vector **b** within the range of values for which **B** is optimal, no change is required to **B**. In this case, there is no change to the shadow prices because there was no change to **B** or **c**. In other words, the shadow prices **p** remain valid over the range of **b** for which **B** is optimal.

Saying that a shadow price is valid over the entire region between two points, including both points, means that the same basis is optimal for both points, and the same shadow prices apply to both. The difference in cost between the two is the shadow price times the difference between the points.

Saying that a shadow price is not valid over the entire region between two points means that different bases are optimal for each point, and the same shadow prices do not apply to both. Therefore, the difference in cost between the two points is not related to a single set of shadow prices. The change in basis between the points disrupts the linear relationship between the costs. The difference in cost between the two points is not the shadow price times the amount of change.

Consider a plot of the objective value against one of the inputs given in Figure 2-1. The optimal objective value with an input of 11 is plotted with the large point in the center. The slope of thick line represents the shadow price for the plotted input. That shadow price is only valid over the range of the input for which the line is thick. Notice that that shadow price is not valid at 12. The point at the right end of the dotted line is at the objective value that would be optimal if the shadow price remained valid up to 12. The true optimal value is also plotted for the input value of 12. This discontinuity between 11 and 12 indicates that there is a discontinuity in the cost function. This signifies a change in basis, so the difference in objective value is not equal to the shadow price times the amount of change. On the other side, the shadow price is valid over the entire range between 10 and 11. In other words, there is no change of basis or discontinuity in the cost function between 10 and 11. Therefore, the difference in cost is simply the value of the shadow price, since there is unit change between 10 and 11.

We define *score differentials* as the difference in objective function value for two points, regardless of the basis for which the point is optimal. In cases where the shadow prices are valid over the entire region between two points, the shadow price is the same as the score differential. In cases where it is not, the shadow price is not the same as the score differential.

2.2 Visualization Techniques

The system described in this thesis presents visualizations of multi-dimensional spaces to help users understand the results of optimizations. Previously, there have been many attempts at the visualization of multi-dimensional spaces. One approach is to use the Retinal Variables of Jacques Bertin [4]. Originally an idea from cartography, these variables allow for the display of three or more dimensions on a two dimensional surface. They include the location, orientation, texture, shape, size, value or darkness, and hue [11] of points and regions. Each of these variables, when matched to a dimension for plotting, increases the number of dimensions that are being visualized.

Jones [7] describes several other approaches to multi-dimensional visualization. For example, projection is another approach to visualizing three or more dimensions in two dimensions. Projection works well for visualizing three dimensions, but it be-

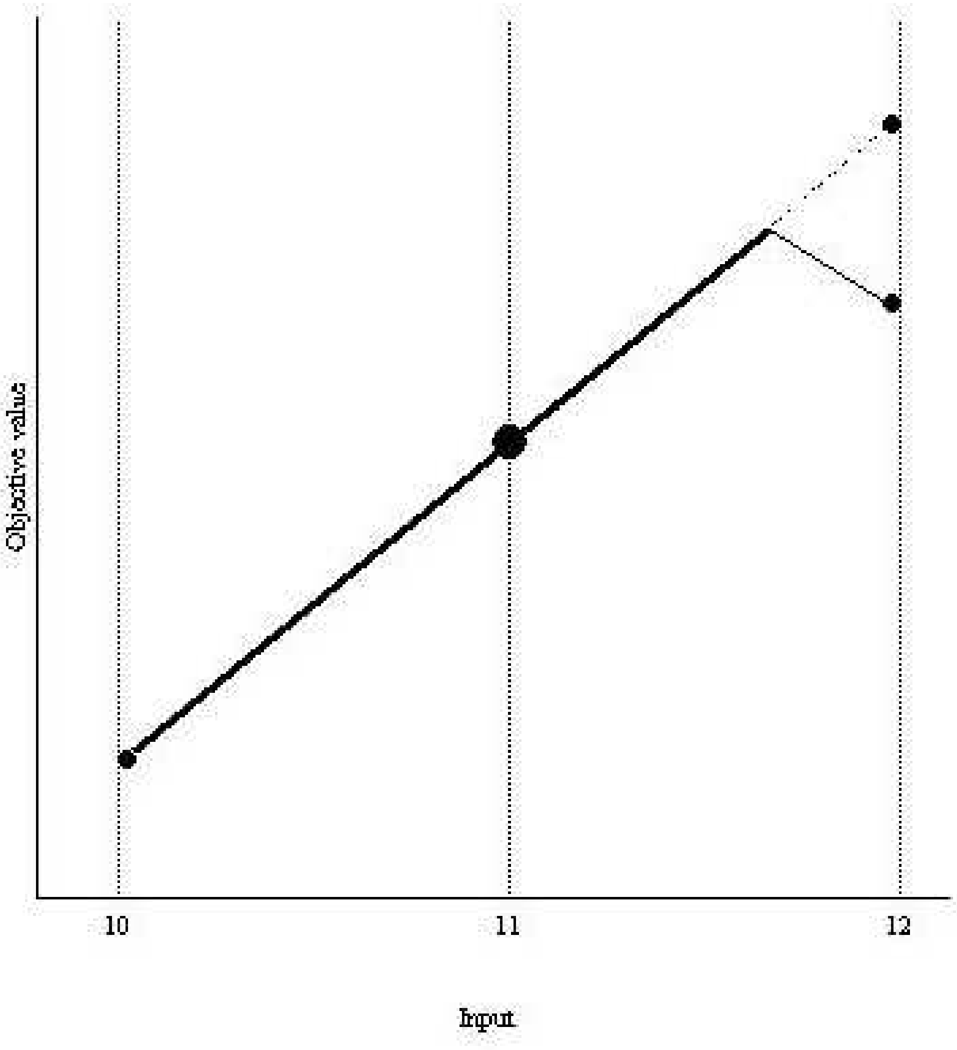


Figure 2-1: Explanation of Shadow Prices

comes very complicated in spaces with higher dimensionality. Another approach is to represent the information in multi-dimensional tables, but this is not very effective for many people because it forces the user to conceptualize the meaning of numbers rather than visually see the relationships between the data. A third approach mentioned by Jones is parallel coordinates, where the axes of each dimension are parallel to one another. A point is plotted by marking the appropriate location on each axis and connecting the marks with lines. This could scale to an arbitrarily high number of dimensions. Finally, Jones presents the idea of "worlds within worlds," where higher dimension spaces are created by nesting two and three dimension spaces within one another.

2.3 HSI/GUI Design

Although the focus of this work has not been on the interface *per se*, but on identifying information about the optimization results that would help a user understand them, we are somewhat aware of the large body of research on user interface design. This section describes some of that work and its relationship to the current work. There are many guidelines and principles that have been proposed for designing and building a user interface. The idea that comes up most often is to understand the different kinds of people who will be using the interface and the task to be supported and to keep those things in mind when designing the system [12]. This will help to ensure that all users find the interface easy to use and powerful enough to support their work. This was a major focus of our work, for providing users with information and tools that they would find useful and informative was a major goal. There are several other principles that were considered during the design and implementation process. For example the idea that behavior and appearance should be consistent [1] is evident in how we use similar methods to present information to the user. Other ideas that generally motivated the design were that the user should have complete control over the interface and that the user should have the ability to reverse actions. Another set of more user-centered principles lists the following as rules of interface design: focus on the users and their tasks and not on the technology, consider function first and presentation later, conform to the users' view of the task, do not complicate the users' task, and deliver information and not just data [6]. These rules were also held in mind as we designed for the user. Finally, a central principle in visual design for presenting large amounts of data to a user is to give an overview first, then zoom and filter, then provide details on demand [12]. This principle is very important to the design of our system, which presents large amounts of information about optimization to the user.

In addition to following these principles, there are design processes that one can follow to help create a successful interface. For example, there is the GUIDE process, which stands for Guide to User Interface Design and Evaluation [10]. The process begins by defining the task and usability requirements. This involves modeling the tasks the users will perform, modeling the objects and information that the user will need when using the system, and defining the interaction style that the users will employ. These models are then used to design the actual interface. Prototype interfaces are created from the design. Evaluation of these prototypes based upon the usability requirements and task definitions can lead to more modeling and further design, or to the decision to implement the prototypes. This process, especially when iterated, ensures that the interface satisfies all the task requirements while supporting the users and their preferred mode of interaction.

Originally, we had planned to design the system presented in this thesis using a process like the GUIDE process. However, it was not possible to get access to the real users of the system. This prohibited us from evaluating the interface with real users and improving it based on those evaluations. We were not able to iterate the process as is suggested and were forced to evaluate our design based primarily on our understanding of the problem and the domain. However, we did review our designs with Chris Nielsen, whose extensive study of our target users is reported in his thesis [9].

2.4 Visualization of Optimization Results

A survey of journals and conferences revealed that little work has been done in visualizing optimization results for the purposes of facilitating user understanding and trust. The following conferences from the past five years were explored:

- International Conference on Intelligent User Interfaces (IUI)
- Conference on Human Factors in Computing Systems (CHI)
- IEEE Symposium on Information Visualization (IEEE Infovis)
- International Conference on Information Visualisation (IV)
- International Conference on Visualization (IEEE Vis)
- Symposium on Visualization (VisSym)
- Institute for Operations Research and the Management Sciences (INFORMS) International
- Military Applications Society Annual International Meeting
- International Workshop on the Integration of AI and OR Techniques
- International Conference on Optimization and Optimal Control
- Military Operations Research Society Symposium

Research regarding the use of optimization for planning has focused on the actual optimization algorithms. The interface to these systems often seems to be an afterthought that only allows the user to do exactly what the system requires him to do. On the other hand, research in visualization and human computer interfaces has not focused on planning systems or logistics and scheduling systems.

We therefore conclude that this attempt to enable user to see into a planning system's results and to compare those results across multiple plans represents a new research area.

Chapter 3

Current Channel Route Network Planning Process

This chapter is based upon Chapter Two of Chris Nielsen's Master of Science Thesis [9]. Refer there for a more detailed treatment.

3.1 Introduction

The *Channel Route Network* is used to move cargo and personnel from the Continental United States to locations around the world. It is a key component of the United States Military logistics system during peacetime. This peacetime logistics system is responsible for ensuring that military personnel have the resources necessary to respond quickly during contingency operations and times of conflict. The channel route network is planned and administered by the United States *Air Mobility Command* (AMC), which is the division of the United States Transportation Command responsible for air transportation. It is their responsibility to ensure that the military is ready for a war effort.

3.2 Problem Domain

This section describes the Channel Route Network (CRN), covering the airports making up the network, the planes flying between the airports, the cargo moving through the network, the air crews flying the planes, and the structure of the plans.

3.2.1 Airports

Airports are the places where cargo is onloaded and offloaded, and where planes takeoff, land, and refuel. Airports that serve as the starting point of a channel route mission are called *Aerial Ports of Embarkation* (APOE), and those that are final destinations are called *Aerial Ports of Debarkation.* There are six APOE's in the Continental United States (CONUS); each one traditionally serves a region of the world. Each airport has resources required to build cargo pallets, transport pallets to and from aircraft, and refuel aircraft. The maximum number of aircraft that can be simultaneously serviced is called the *Working Maximum on Ground* (WMOG), and the maximum number that can be at an airport at one time is called the *Parking Maximum on Ground* (PMOG). In addition, some airports have set operating hours and quiet hours. Some airports are also hubs, similar to the hubs used by many major airlines and cargo shippers. Each airport is referred to by a three letter abbreviation; the abbreviations and the airports they signify are listed in Appendix A.

3.2.2 Planes

There are four types of planes available to fly channel route network missions; they are based at four different airports. The characteristics of the different plane types are summarized in Table 3.1. Each plane type has a true capacity and a planning capacity. The true capacity is the maximum amount of cargo that the plane can physically carry, and the planning capacity is the maximum amount of cargo that a planner will assign the plane to carry. The C-5 and C-17 are the largest of the plane types; they can carry very large cargo, such as tanks and helicopters. All but the C-130 can refuel in flight, which greatly increases the distance a plane can fly.

Type	**Origin**	**True Capacity** *(tons)*	**Planning Capacity** *(tons)*	**Operating Cost** *($/hr)*
C-5	DOV	145	50	19000
C-17	CHS,RMS	85	25	9000
C-141	WRI	34	16	8900
C-130	RMS	22	8	6000

Table 3.1: Characteristics of planes flying channel route missions

When there are not enough military aircraft available to fly all the channel route missions, commercial aircraft are used in two ways. First, commercial carriers have pledged a certain percentage of their aircraft fleet and crews to what is called the Civil Reserve Air Fleet (CRAF). These resources will be used for military reasons when the president activates the CRAF program. Second, the military enters into contracts with commercial carriers to handle the channel route missions that exceed it's current capability.

3.2.3 Cargo

There are 27 different kinds of cargo that are delivered through the channel route network, ranging from aircraft parts to mail. These cargo fit into three basic size categories: bulk, oversized, and outsized. Bulk cargo fits on a standard sized pallet, which will fit on any aircraft type. Cargo that cannot fit on a single pallet is called oversized cargo. Multiple pallets are combined to ship oversized cargo. Finally, outsized cargo is very large and unwieldy, like tanks or helicopters. 99% of the cargo shipped through the channel route network is in the bulk category. All cargo is also assigned a priority that determines the order and manner in which it is shipped. For example, high priority cargo will go out first via plane, whereas very low priority cargo will be shipped later, perhaps by truck or ship.

The majority of the cargo shipped throughout the network begins in the Continental United States (CONUS). This cargo starts at a distribution warehouse and moves to one of the output airports via truck. There the cargo is loaded onto pallets, and then on to planes. The cargo is then flown to its destination, either directly or through one or more enroute airports. Once cargo reaches its destination, it is put into a holding facility, from which it is loaded onto a truck that will take it to the customer. Cargo may be offloaded from a plane at an intermediate destination before it reaches its destination if that intermediate airport is a hub. The offloaded cargo is later loaded onto another plane destined for the cargo's final location.

3.2.4 Flight Crews

Channel route network missions also provide valuable training to pilots to maintain their wartime readiness. In flying channel route missions, pilots gain the required flying experience that is mandated by the flying hours program. This program says that pilots must fly a minimum number of hours before they are considered to be trained. Channel route missions provide a predictable and recurring means of meeting these requirements.

Another constraint on the system is the *crew duty day* (CDD) limit, which is the number of consecutive hours a crew can fly. The crew duty day starts when the crew arrives at the origin airport before takeoff, and ends when the crew arrives at the destination airport. The crew is required to rest for a minimum amount of time between duty days. There are two types of crews, *basic crews* and *augmented crews*. The augmented crew consists of two basic crews, which allows the aircrew to extend its CDD. The number of air crews available to fly channel route missions also constrains the planning system.

3.2.5 Plan Structure

There is a hierarchic structure to the plans created for the channel route network. At the lowest level, a plan is comprised of flight legs. A *flight leg* is a flight between two airports by a plane carrying some cargo. A set of flight legs makes up a *channel route mission*. A mission is flown by a single plane making stops at one or more airports along the way. At each airport it may pick up or drop off cargo. All missions end where they started. A set of missions is collectively called the *channel route network* *plan*. All of the missions in the plan must collectively ship all of the cargo around the network, and return the planes to their originating airports.

3.3 Current Manual Process

Creating channel route network plans is a lengthy, manual process. It is further complicated by the fact that there are other, higher priority, uses for planes. As a result, planes are often taken away from channel route missions to fly in other situations, such as contingencies and training exercises. These planes can be taken at any time during the planning and execution process, and the planners must react to the changes.

Currently, plans are generated one month at a time. Planning for a particular month occurs in the two months before that month. These two months are called the *planning period*, and the month for which the plan is created is called the *execution month*. During the planning period, channel route planners create all the channel route missions for the execution month. All missions are created fifteen to thirty days before the beginning of the execution month. After this initial creation, changes continue through the remainder of the planning period and the execution month.

The planning process involves multiple people working together. To begin, a person called the *organic channel scheduler*, who is the main planner, accesses the channel route schedule for the previous month and makes any modifications to deal with any known special circumstances, such as runway construction. At this point, the plan is called the *initial cut*.

The plan is then passed to two other people, called the *barrelmaster* and the *cargo bookie*. The barrelmaster determines if there are enough aircraft available to fly all of the missions in the initial cut. If not, he will either drop or modify lower priority missions. These dropped missions may still be flown, but with commercial aircraft. Every time that there is a change in the number of planes allocated to flying channel route missions, the barrelmaster performs this check of whether all missions can still be flown.

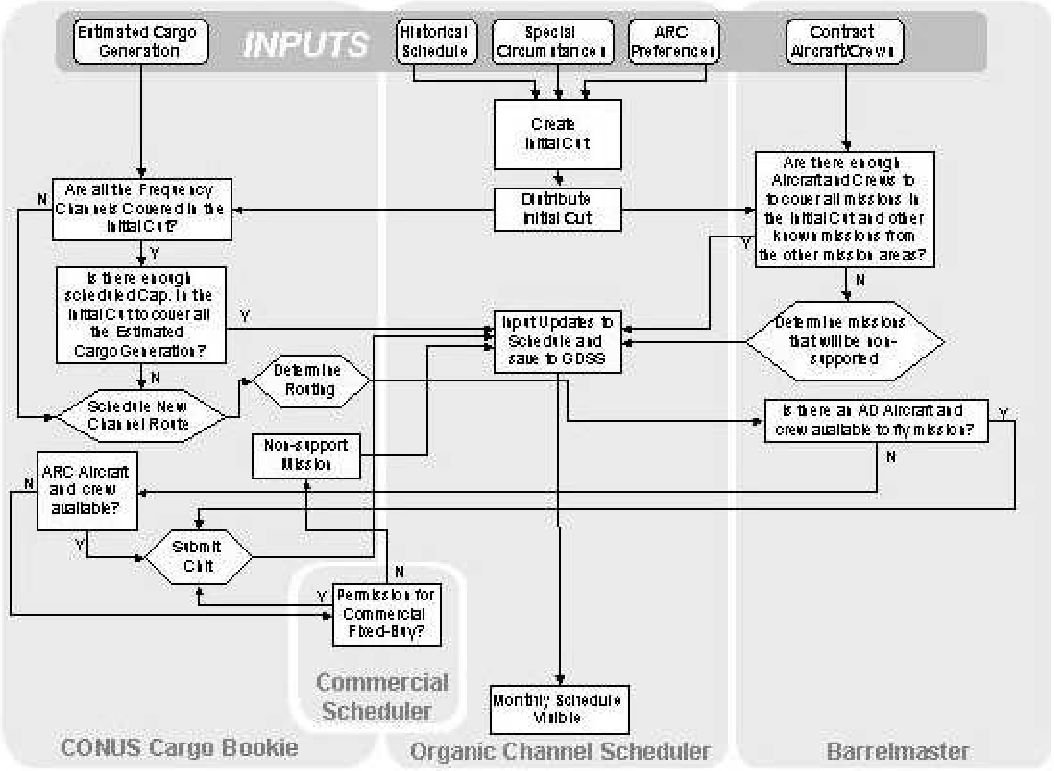


Figure 3-1: The current channel route planning process (taken from [9])

The initial cut is also passed to the cargo bookie. His job is to determine if there are enough scheduled missions to move all the required cargo. If not, he requests the organic channel scheduler to create new channel route missions. Each new mission will be covered by a military aircraft if possible, and a commercial aircraft otherwise. Either organic channel scheduler or the barrelmaster is responsible for scheduling this mission. Figure 3-1 provides a snapshot of the current manual planning process, as described in [9].

3.4 Automated Optimization

The *Channel Route Network Planning System* (CRNPS) [9] is the first step to automating the entire channel route planning process. It automates the creation of the initial cut used by the barrelmaster and the cargo bookie. Each run of the CRNPS consists of three parts: setting up the optimization formulation, solving the optimization, and processing the results of the optimization into the initial cut. In the first step, a technique called *composite variable formulation* is used to generate the actual optimization formulation to be solved, based on the various input parameters and the cargo to be shipped throughout the CRN. There are several possible objectives of this formulation (Table 3.2). Only one objective can be optimized at a time, and in most of cases, the operating cost is minimized. The constraints of the formulation are:

- the amount of cargo on a flight leg must be less than or equal to the capacity of the plane flying the flight leg,
- an aircraft can only leave airports at which it previously arrived,
- cargo can only leave airports at which it previously arrived,
- the number of aircraft used must be less than the number available,
- the number of required flying hours must be met for each plane type,
- the WMOG and operating hours at each airport must be honored,
- all cargo must be flown to its destination, and
- an aircraft must start and end its channel route mission at its home base.

This optimization formulation is an example of a *mixed integer program*, where some of the decision variables must take integer values in the optimal solution. In this case, the number of planes used must be an integer. The other decision variables can take positive real values in the optimal solution.

The next step of the CRNPS is to solve the optimization using XpressMP , a commercial software package for solving optimizations. Finally, the optimal values of the decision variables, as determined by XpressMP, are translated into a number of output files that specify the initial cut (Table 3.3). An excerpt from the output file that lists the missions in the initial cut, *missions_in_solution.out*, is given in Figure 3-2. The initial cut created by the CRNPS can then be evaluated and used by the barrelmaster and the cargo bookie in the same way as in the manual process.

Objective (minimize)	**Description**
Aircraft	The total number of aircraft used
Operating Cost	The monthly aircraft operating costs of the channel route network
Missions Flown	The total number of channel route missions flown
Unused aircraft capacity	The total amount of unused aircraft capacity over the CONUS outbound flight legs (i.e. Transportation Working Capital Fund (TWCF) Utilization Rate)

Table 3.2: Potential objective functions to be minimized

File Name	**Description**
Commodity_times.out	Lists the cargo to be carried around the network
Flt_leg_ute_rates.out	Lists the utilization rates of each flight leg
mission_statistics.out	Lists several mission statistics, including flying hours
missions_in_solution.out	Lists each mission, specifying flight legs and cargo carried
TWCF_Ute_Rates.out	Lists TWCF utilization rates for each mission
wrap_arcs.out	Lists number of each plane type used
probabilities.dat	Lists the probabilities of having certain amounts of resources available.

Table 3.3: Outputs of the optimization

```
------------------------------------------------------------------------
Mission = 41
Column = 3355
Value in Optimal Solution = 1
SRCM Startday = 11
Derived from Composite Route 2
Aircraft:        C17-CHS
Crew Type:       1
Sequence:        Base   ArrDay ArrTime  CR?  DepDay DepTime
                 ====   ====== =======  ===  ====== =======
                 CHS     11          0    n     11          0
                 DOV     11        1.5    n     11       4.75
                 AVB     11      13.65    Y     12        9.4
                 CHS     12       20.7    n     13          0
Flying Hours:    21.7
Grnd & CR Time:  23
TAFB:            44.7

Potential OD-Covers: DOV->AVB(10.1016), DOV->AVB(8.7159), DOV->AVB(4.24277),
Freq Reqt Covers:
Cargo Loading:
--------------
Loading # 0: DOV->AVB(# 37, ALT=9, tons=10.1016),
DOV->AVB(# 38, ALT=10, tons=8.7159), DOV->AVB(# 39, ALT=11, tons=4.24277),
Tons = 23.0603
------------------------------------------------------------------------
```

Figure 3-2: Sample mission from *missions_in_solution.out*

```
PHT=1,2
Fract=0.2,0.1
SF=1,3
Seq=Regional Tours,Direct,Hub-and-Spoke
# 3 per aircraft type:
C5-DOV=11,13,12
C17-CHS=14,16,15
C141-WRI=3,5,4
C17-RMS=2,4,3
C130-RMS=14,16,15
allCombinations=y
```

Figure 3-3: File specifying possible values for optimization inputs

The CRNPS can be run many times on many different sets of input values, creating a plan for each one. This is done using what is called the input perturbator. The first step to running the perturbator is to specify the possible values for each of the inputs. Figure 3-3 shows an example file listing the possible values for the inputs. The possible values for a single input are listed on a line in the file and are separated by commas. The name of the input begins the line. Lines beginning with a # are ignored. The perturbator reads this file and runs the optimization once for each possible combination of input values. The perturbator saves the output from each of those runs in a directory whose name is a combination of all the parameter values used in that run. For example, if a run of the CRNPS with the first possible value of each of the inputs listed in Figure 3-3, the output would be saved in the following directory:

root/SolvedScenarios/PHT1_Fract2_SF1_SeqRT/Cost/11_14_3_2_14

where root is the directory from which the optimization is run.

Chapter 4

Design of the System

4.1 Introduction

The goal of this research is to reveal information needed to understand optimization solutions in order to make those optimization systems more transparent. One such system is the Channel Route Network Planning System (CRNPS), described in Chapter 3. This chapter presents the design of the Channel Route Information System, or ChRIS, which provides the desired transparency and supports understanding of the system's operation and solution.

When using ChRIS, the user is trying to select a plan for execution out of the set of plans created by the CRNPS. ChRIS helps the user to do this by revealing information needed to understand and compare the plans created by the CRNPS. There are two types of tools in ChRIS: tools for comparing multiple plans, and tools for visualizing a single plan. The tools that compare multiple plans along many dimensions allow the user to compare the general characteristics of each plan and allow him to select the overall best one, according to the selection criteria he imposes on the problem. The plan visualization tools show the user information about the specifics of a plan. This further information should aid the user in making his decision about which plan to use.

This chapter has three major sections. The first section explains what steps must occur before interaction with ChRIS may begin and what information these steps make available. In the next section, the tools for comparing multiple plans are presented, and in the third section, the tools for visualizing a single plan are discussed. For each of the tools discussed in these two sections, we discuss why it is necessary, and what it looks like. We also explain what information the user can get out of the tool, and how to get that information out of the tool.

In this work, we have been more interested in discovering and revealing the most important information about the optimization, and less concerned with usability issues of how that information is presented to the user. Although we believe usability is important for this work to be helpful to real users, at this stage of the research, the primary issue was determining what information is required to understand optimization solutions and outputs.

4.2 Preparation for System Use

There are several steps that must occur prior to interacting with ChRIS, the Channel Route Information System. The first step is to run the input perturbator (see Section 3.4), which runs the optimization many times, using various numbers of resources. Each run of the optimization produces a set of output files that fully specify the plan it has created (Table 3.3). The most important output file from the optimization is the one listing the missions in the plan. This tells which planes are flying between which airports carrying what cargo. There are also separate listings of how many planes are used, how long each type of plane is flying, and what cargo is being moved.

Next, the output of the input perturbator must be processed into a format that is usable by ChRIS (see Section 6.4.1 and Section 6.4.2). This processed information, as well as some unprocessed information directly from the optimization, is presented to the user using the displays described below.

The final step that must occur before use of ChRIS may begin is to define the probabilities of having each amount of resource available. For each of the possible input values, a probability of availability must be given. These probabilities are

```
11 .3 13 .5 12 .2 #C5-DOV
14 .6 16 .2  15 .2 #C17-CHS
2 .6 4 .3 3 .1 #C17-RMS
14 .2 16 .6 15 .2 #C130-RMS
no3 .8 5 .1 4 .1 #C141-WRI
1 .33 2 .67 #PHT
0.1 .75 0.2 .25 #Fract
1 .625 3 .375 #SF
D .33 HS .33 RT .34 #Seq
```

Figure 4-1: Example file specifying probability of resource availability

not created by the optimization and could be specified manually or generated from historical data. Figure 4-1 shows an example file listing these probabilities. Each line in the file represents one of the inputs. There are n sets of two numbers in a line for an input that has n possible values. The first number of each set is the actual value, and the second is the probability of having that number available. For example, the file in Figure 4-1 says that there is a .3 chance of having 11 C5-DOV available. The probabilities for each of the possible values for a single input must sum to one. Under an independence assumption, the probability of having a set of resources is simply the product of the probabilities of having each individual resource. For example, from the figure the probability of having 11 C5-DOV available is .3, and the probability of having 14 C17-CHS available is .6, then the probability of having 15 c5-DOV and 14 C17-CHS available is .18. Once the perturbator has been run, the output has been process, and the probabilities have been specified, ChRIS is used to explore the set of plans, enabling the user to select the best one.

4.3 Comparing Multiple Plans

The data produced by the planner (see Figure 3-2), when it is run with each amount of potentially available resources, does not easily allow for human comparison of large numbers of plans. To support comparisons between plans, a set of graphical tools was developed (Figure 4-2). The main component of this tool set is a *Five-Dimensional Graph*, and the other parts of the tool set facilitate the display and use of the graph.

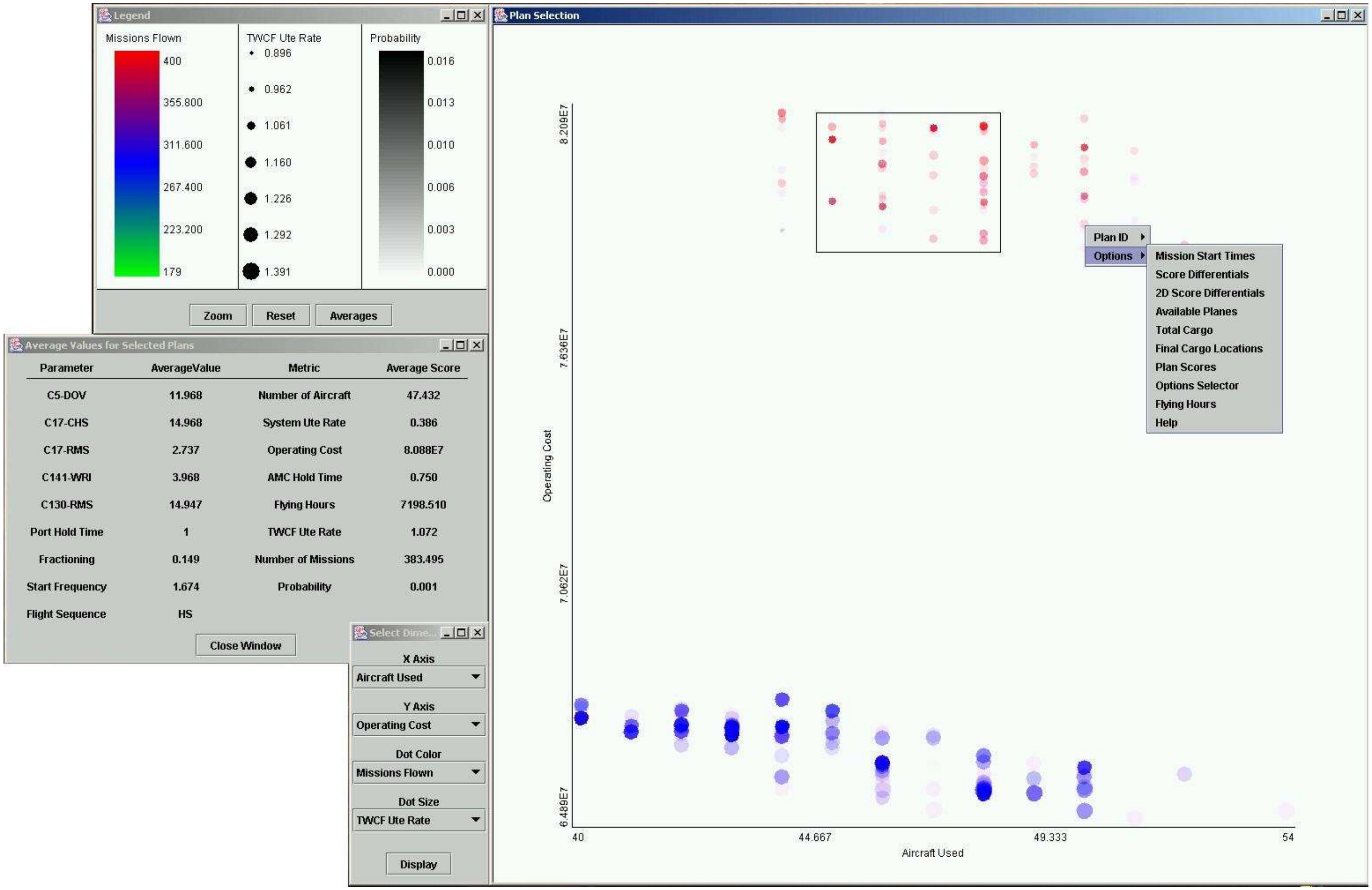
Metric	**Description**	**Objective**
Aircraft Used	The total number of aircraft used to operate the channel route network	minimize
System Ute Rate	Average utilization rate of aircraft over all flight legs in the plan	maximize
Operating Cost	The total operating cost over all channel route missions	minimize
AMC Hold Time	The average time between when the cargo is ready for shipping and when it is delivered, over all cargo commodities	minimize
Flying Hours	The total number of flying hours provided by the plan	fit within a range
TWCF Ute Rate	Average utilization rate of aircraft over all CONUS outbound flight legs in the plan	maximize
Missions Flown	The total number of channel route missions flown in the plan	minimize

Table 4.1: Descriptions of the seven metrics used to score a plan

These include a tool that allows the user to specify what information is plotted along the dimensions of the graph, a legend for the non-Cartesian axes of the graph, a pop up menu that leads to more detailed information about a specific plan, the ability to zoom in on regions of the graph, and a display of average values for a set of selected plans.

4.3.1 Five-Dimensional Graph

Each of the plans created by the CRNPS has its own merits and drawbacks, and it can be difficult for a user to distinguish and compare many plans. We use the *Five-Dimensional Graph* as a visualization tool for multiple plans. It shows each plan as a dot in a five-dimensional space. The first two dimensions of the graph are the x and y axes, and the third and fourth dimensions are point size and color (Figure 4-3). These four dimensions of the graph show four of the seven possible metrics used to evaluate a plan (described in Table 4.1). Which four metrics to show are selected using a *Metric Selection Tool* (Figure 4-4).



Parameter	AverageValue	Metric	Average Score
C5-DOV	11.968	Number of Aircraft	47.432
C17-CHS	14.968	System Ute Rate	0.386
C17-RMS	2.737	Operating Cost	8.088E7
C141-WRI	3.968	AMC Hold Time	0.750
C130-RMS	14.947	Flying Hours	7198.510
Port Hold Time	1	TWCF Ute Rate	1.072
Fractioning	0.149	Number of Missions	383.495
Start Frequency	1.674	Probability	0.001
Flight Sequence	HS		

X Axis: Aircraft Used
Y Axis: Operating Cost
Dot Color: Missions Flown
Dot Size: TWCF Ute Rate

Figure 4-2: Plan comparison tool set. The *Five-Dimensional Graph* is shown on the right, with a set of plans selected and with the *Pop Up Menu* displayed for a plan. The three smaller windows on the left are, from top to bottom, the legend for the three non-Cartesian axes, the display of average metric scores and parameter values for the selected plans, and the *Metric Selection Tool*

The fifth dimension is the transparency of the dot. Transparency is one of multiple ways of visualizing uncertainty [8]. The transparency of a point indicates the likely availability of the resources used to create the plan represented by that point. A less transparent dot means that it is more likely that that set of resources will be available during execution. An example of the *Five-Dimensional Graph* is given in Figure 4-3, and its corresponding *Legend* is given in Figure 4-6.

The user is also able to select a region of plans for further investigation. By clicking and dragging the mouse over the graph, the user selects a rectangular region of the graph. The beginning and end points of the drag define opposite corners of the rectangle. The user can then zoom in on this region (Section 4.3.3.2), or he can find out the average metric scores and parameter values for the plans within the region (Section 4-9).

Choosing a plan is a process of optimizing seven metrics. For each metric, there is an objective (Table 4.1) that imposes an ordering on the plans. The *Five-Dimensional Graph* (Figure 4-3) is the key tool used to explore the resulting set of plans and their metric scores. Four of the metrics can be plotted on the four variable dimensions of the graph (the fifth dimension, probability, always appears in the graph). The user can customize the graph by selecting the appropriate metric for each variable dimension. Based on the four metrics chosen, the user should look for a certain type of dot in a certain region of the graph. If the objective of all four plotted metrics is minimization, then the user should look for small green dots in the lower left of the graph. Likewise, if he is trying to minimize the metrics corresponding to the x axis and dot color and maximize the metrics corresponding to the y axis and dot size, he should look for big green dots in the upper left of the graph.

The fifth dimension represents the probability of having available the set of resources used to create that plan. Probability indicates two things. First, it tells the user how much attention to pay to that plan. A low probability, represented by a nearly transparent dot, indicates that a plan should probably not be used, since it is unlikely that that set of resources will be available. Therefore, the user probably should not invest much time exploring its details.

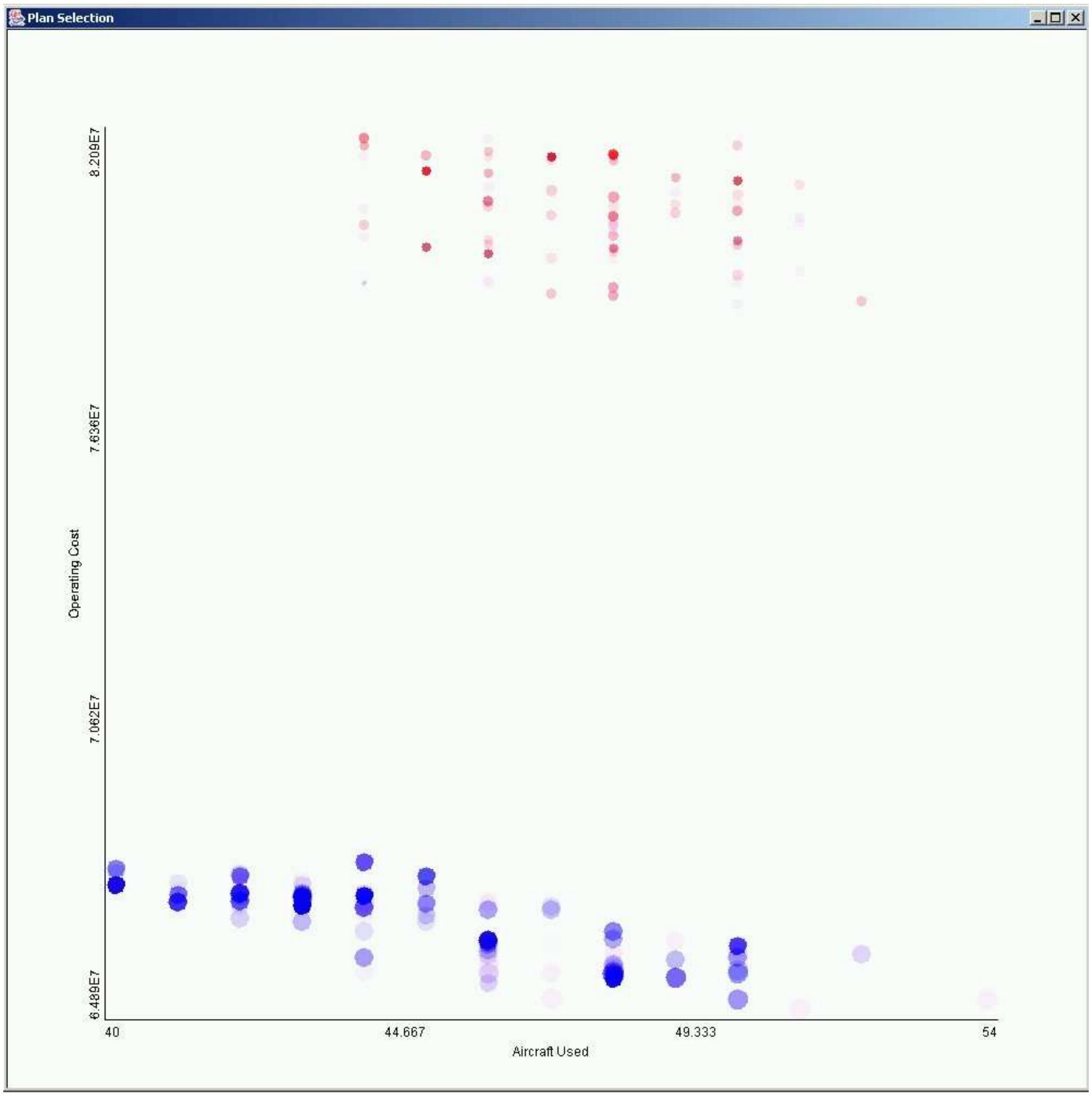


Figure 4-3: *Five-Dimensional Graph* Display (see *Legend* in Figure 4-6)

Second, it indicates the possible need for robustness in the plan. We define robustness as independence from external effects, that change which resources are available. If the plan has a very high probability, there is not much need for robustness, because the chance of being forced to change to a different set of resources is relatively low. Likewise, a plan with a low probability would need considerable robustness, for it is likely that the set of resources and, consequently, the plan will change. We discuss in Section 4.4.4.1 how ChRIS supports the selection of robust plans.

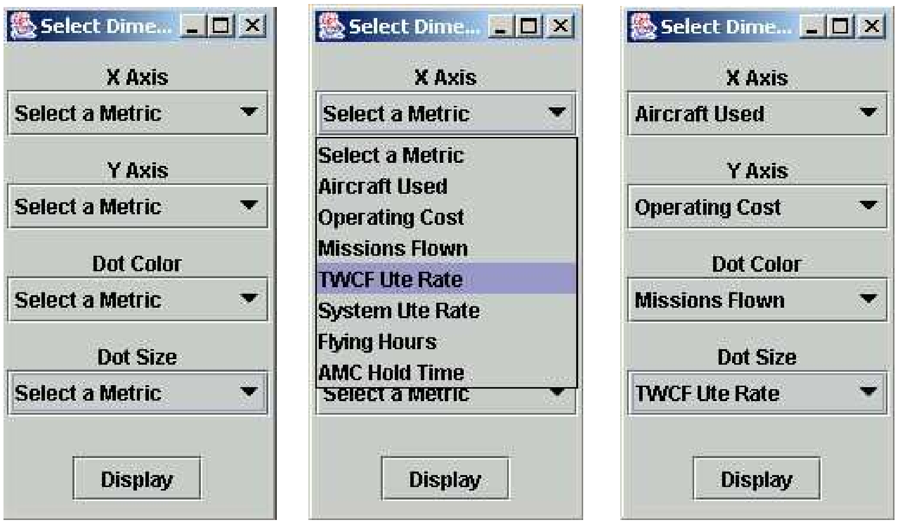
The *Five-Dimensional Graph* is useful because it allows the user to consider information about all of the possible plans at once. It provides several pieces of information about each plan in a compact way that allows for easy human examination.

4.3.2 Metric Selection Tool

The *Metric Selection Tool* is a small window with four drop-down selection boxes and a single button. Each of the selection boxes corresponds to one of the first four dimensions of the *Five-Dimensional Graph* (Figure 4-4), and it is used to choose the metric to be plotted along that dimension. A metric must be selected for each of the four dimensions, and no metric can be selected multiple times. If the user tries to display a graph when either of these is false, he is notified; otherwise, the corresponding five dimensional graph is displayed, with each of the available plans plotted appropriately.

The *Metric Selection Tool* (Figure 4-4) gives the user the ability to format the information presented on the five-dimensional graph in a way that he chooses. He can format the graph depending on what information and which dimensions he thinks are most important. For example, if his most important considerations are Operating Cost and TWCF Utilization Rate, and the most important dimensions to him are the x and y axes, then he will match up those metrics to those dimensions.

The *Metric Selection Tool* also allows the user to view multiple graphs for the same set of plans by choosing different combinations of metrics. There are $\begin{pmatrix} 7 \\ 4 \end{pmatrix}$, or 840, possible matchings of metrics to dimensions, and a distinct graph can be



X Axis: Aircraft Used
Y Axis: Operating Cost
Dot Color: Missions Flown
Dot Size: TWCF Ute Rate

(a) (b) (c)

Figure 4-4: *Metric Selection Tool* (a) before any selections have been made, (b) during a selection, and (c) after all selections have been made

created for each one. The user can use the *Metric Selection Tool* to view multiple combinations of metrics until he finds a suitable one. He can also use it to look at information on a certain set of four metrics in multiple ways by changing the axes to which each of the metrics are assigned.

The *Metric Selection Tool* is important because it allows different views of the plan space. Since there are more metrics than there are available dimensions for plotting, the user is allowed to choose the information to view, and he is given some freedom in how the information is displayed. He can also create multiple instances of graphs and look at the same information in multiple ways by selecting and viewing different combinations of metrics for the four variable dimensions in the various graph instances. The graphs are tiled across the screen so that they are all visible at once (Figure 4-5).

4.3.3 Legend of Five-Dimensional Graph

There are three parts of the *Legend* of a *Five-Dimensional Graph* (Figure 4-6). The main part provides information about the color and size dimensions of each five

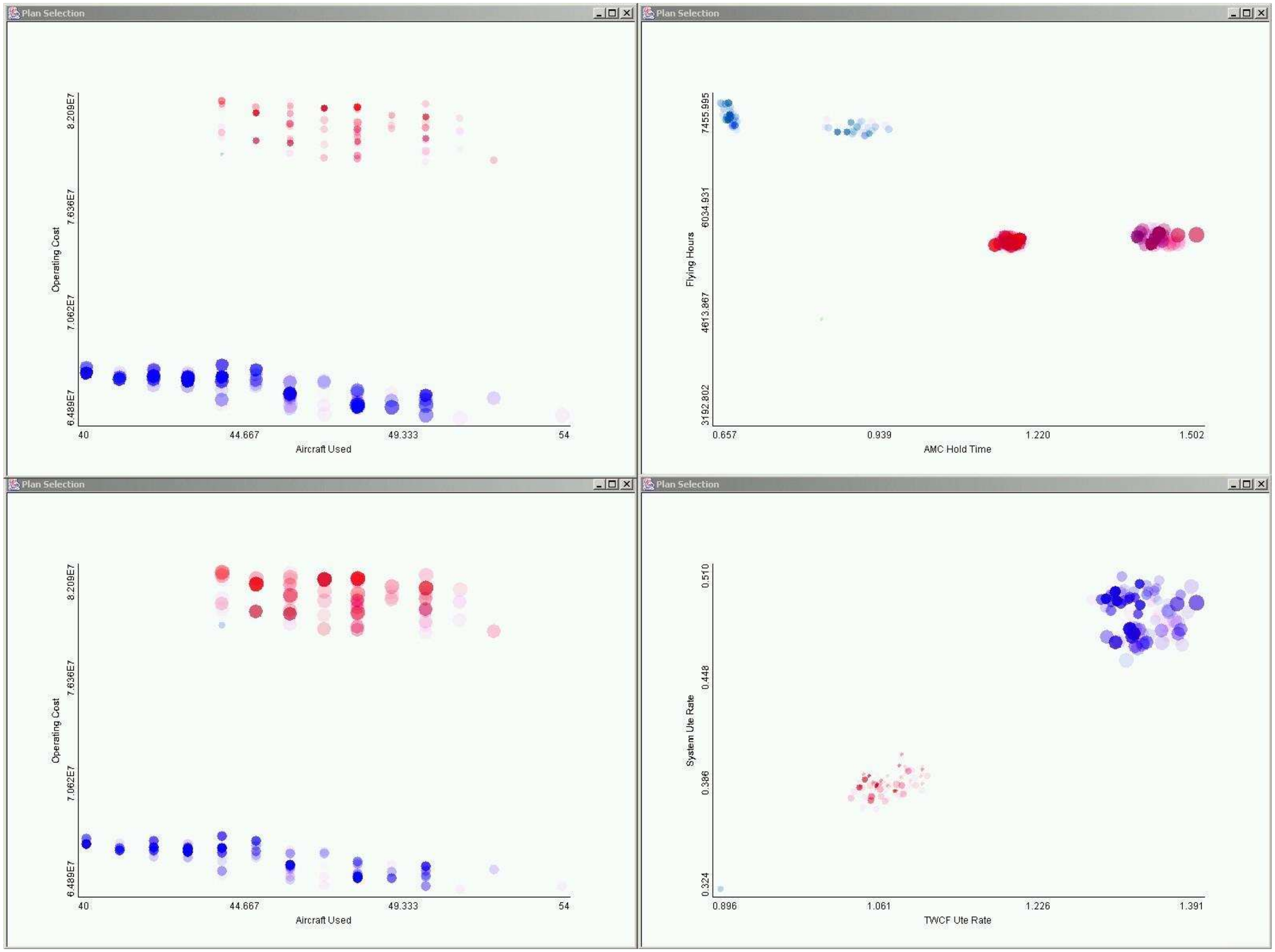


Figure 4-5: Multiple *Five-Dimensional Graphs* visible at once

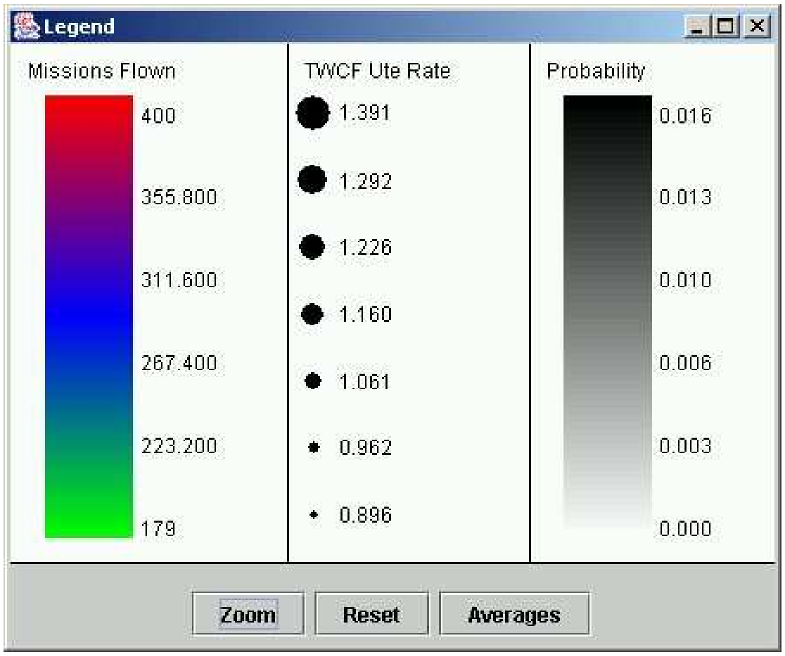


Figure 4-6: *Legend* of the *Five-Dimensional Graph*

dimensional graph. It also provides information about the probability of resource availability for each plan shown on the graph (Section 4.3.3.1). The second part consists of two buttons, labeled "Zoom" and "Reset", that allow the user to zoom in on and reset the graph (Section 4.3.3.2). The third part consists of a button labeled "Averages" that, when clicked, brings up a display of average metric scores and parameter values for a selected region of the graph (Section 4.3.3.3).

4.3.3.1 Non-Cartesian Axis Scales

The main part of the *Legend* consists of three adjacent sections, left to right. The first section contains the color scale and the metric selected to be plotted using color (e.g. Missions Flown). Below this label is a color bar that shows how the color changes from a low metric value to a high value. In the graph, green corresponds to a low value, blue to a medium value, and red to a high value. The color bar is labeled with scores. The maximum and minimum labels give the range of scores for that particular metric. For example, the color scale of Figure 4-6 shows that the number of Missions Flown varies from 179 (green) to 400 (red), for a total range of 221.

The second section of this part of the *Legend* shows the dot size scale, similarly indicating the metric plotted, and giving a series of dots of decreasing size with corresponding metric values. Again, this helps the user to understand the actual metric score for a particular dot size, and the maximum and minimum labels give the range of scores for that metric. The dot size scale of Figure 4-6 shows that the TWCF Ute Rate varies from .0896 (small dot) to 1.391 (large dot).

The third section of the main part of the *Legend* provides information about darkness of dots, showing the resource availabilities as different levels of darkness or transparency. In the graph, plans whose resources are less likely to be available are indicated by lighter dots. Again, the maximum and minimum labels define the range of probabilities. The example in Figure 4-6 says that the probabilities range from 0.000 to 0.016.

The main part of the *Legend* is used to translate the color, size, and transparency dimensions of the *Five-Dimensional Graph* into concrete numeric values. The user can consult the information to decide which plans to consider further. Consider the example in Figure 4-6, where Missions Flown is plotted using color, and that the values range from 179 to 400. Imagine that the user decides to consider only those points that have a value less than 300. From the *Legend* he can see that he should only look at points that are colored between blue and green. It can also be used in the more straightforward sense, where the user finds the actual value of the metric from the appropriate scale in the *Legend*. In other words, he could look up in the *Legend* given in Figure 4-6 the color and size of a large blue dot on the graph to see that the corresponding plan has 300 Missions Flown and a TWCF Utilization Rate of 1.39.

The main part of the *Legend* is necessary because it provides information crucial to understanding the graph. It interprets the color and size values as numeric scores for the corresponding metrics. It is useful to those users who are more interested in the actual metric scores rather than the relative magnitudes, which can be determined by differences in color and dot size on the graph itself. Seeing the range of scores is important because it indicates the sensitivity of color or size to changes in score or size. For example, a wide variation in color or size may correspond to a very small variation in actual score. If this is true, then information for that metric may not be particularly useful, since all plans would fall into a fairly narrow range of values. On the other hand, a small variation in color or size could correspond to a very large difference in actual score, which could be very significant, and the *Legend* allows the user to see if this is the case.

4.3.3.2 Zooming and Resetting Five-Dimensional Graph

On the *Legend* (Figure 4-6), there are two buttons labeled “Zoom” and “Reset.” Clicking and dragging the mouse over the graph draws a rectangle (Figure 4-7). Subsequently clicking “Zoom” expands that region of the graph, allowing the user to get a closer look (Figure 4-8). The color and size scales on the *Legend* are set to fit the selected data, so when the graph is zoomed in, the same plan’s dot may appear with a different color and dot size in the zoomed view. Zooming may be done multiple times in succession, each time drawing a rectangle and clicking the button. At any time, while the graph is zoomed, clicking the “Reset” button will restore the graph to its original state, with all plans visible (e.g. transitioning from Figure 4-8 to Figure 4-3).

The ability to zoom in on the graph (Figures 4-7 and 4-8) allows the user to navigate regions of interest in the graph and is most useful in two situations. The first is when there is clustering on the graph, such as in Figure 4-3. Zooming in on a cluster allows the user to get a better view of the cluster. Spreading the cluster out allows the user to separate the points and consider them individually. It may also make points visible that were not visible before, due to the overlapping of points.

The second situation in which zooming is beneficial is when the user wants to focus on a region of the graph. For example, the metrics chosen could dictate that the user wants to focus on the lower right quadrant of the graph. The user can then zoom in on this region, eliminating all of the plans outside the region that he is no longer considering.

Repeated applications of zooming can be used when one zoom is not enough. For

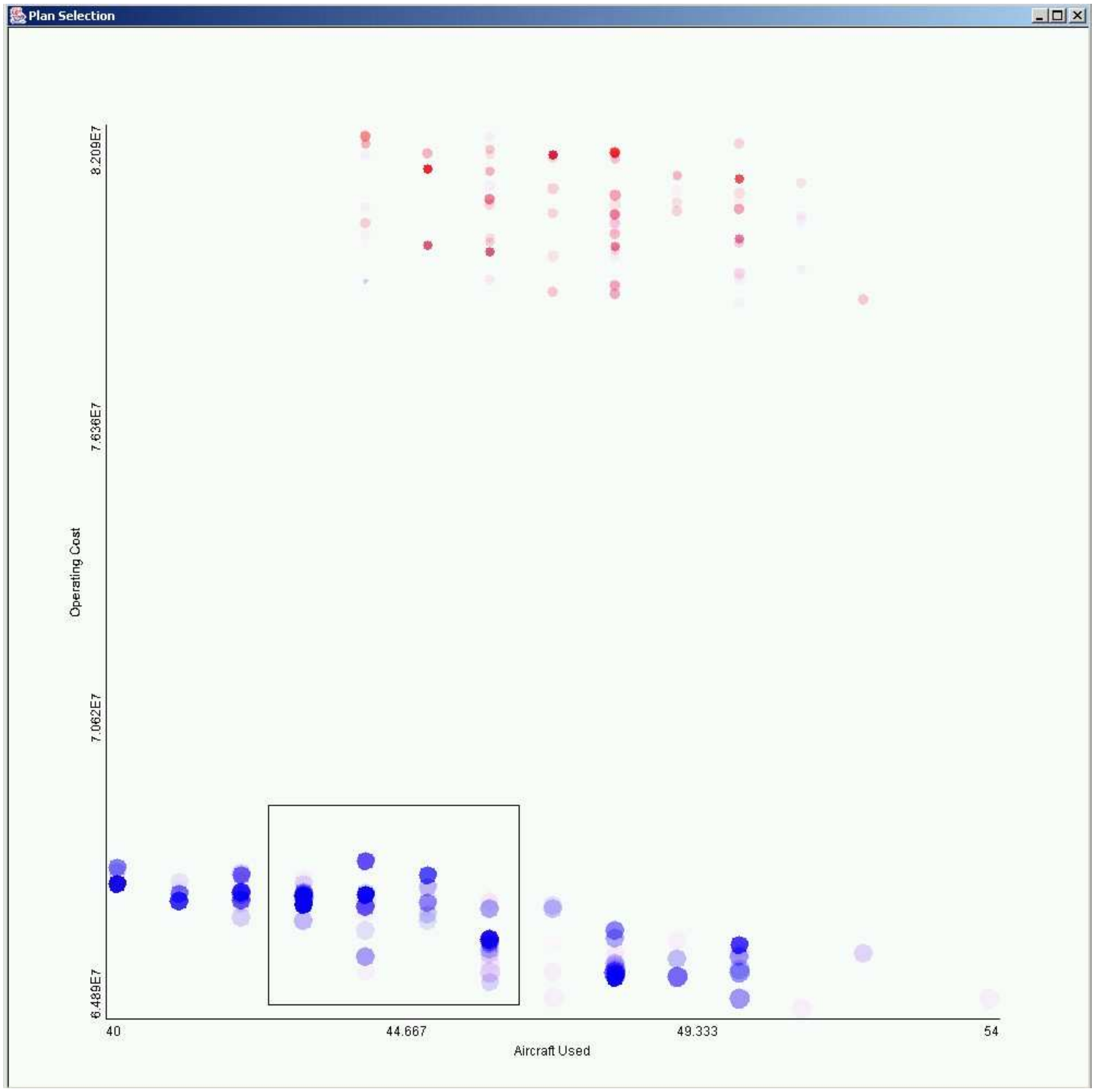


Figure 4-7: *Five-Dimensional Graph* with rectangular region selected

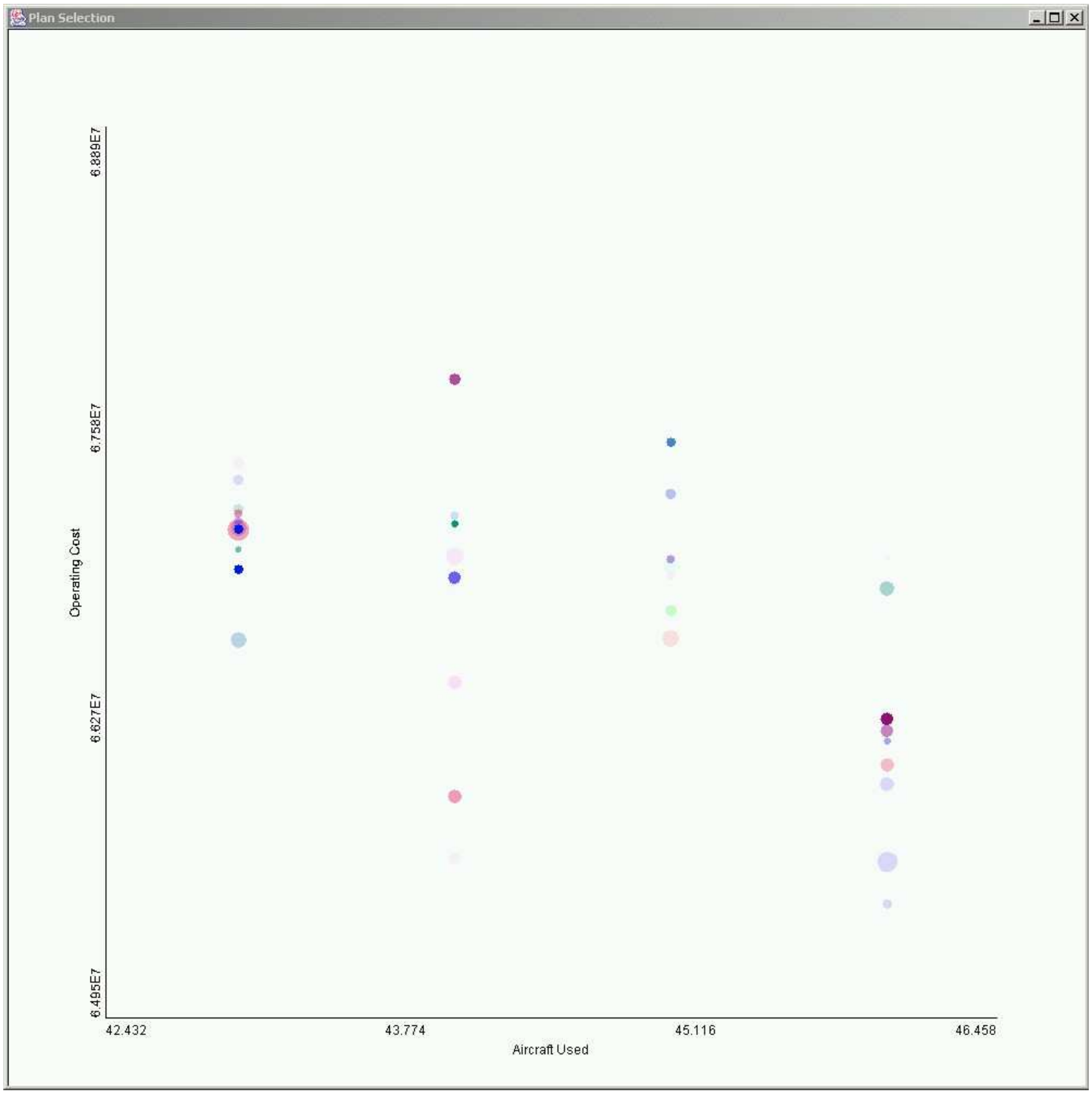


Figure 4-8: *Five-Dimensional Graph* zoomed in on selected region of Figure 4-7

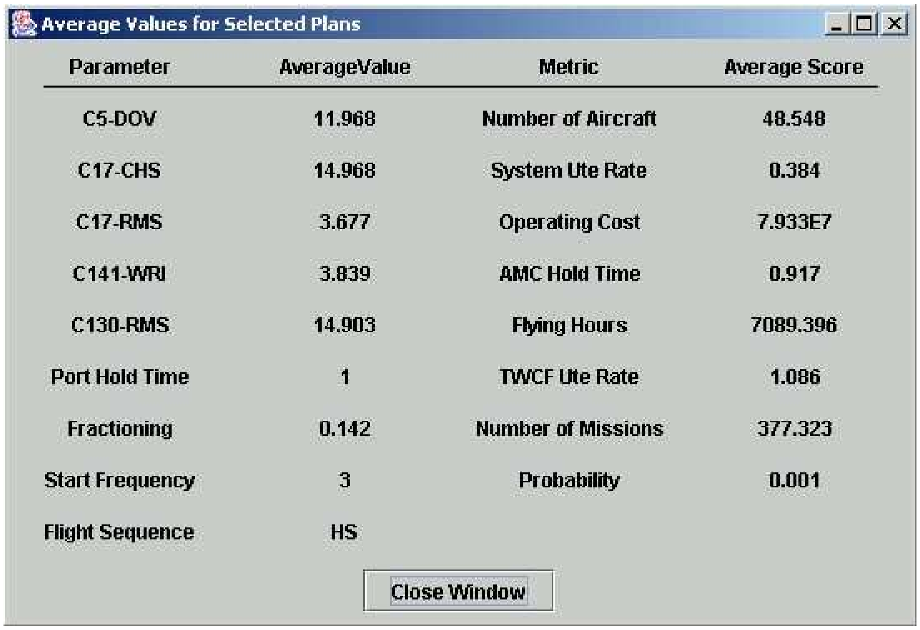
example, there may be sub-clusters within clusters that only become visible after zooming in on the main cluster. Zooming in again allows the user to focus on one of these subclusters from a zoomed in view, rather than from the initial view, which may not be possible. The only way to select the small subcluster from the initial view would be to guess.

Resetting the graph is the only way for the user to zoom out again and focus his attention on another cluster or region of the graph. It provides an easy way to focus on the total plan space after having moved in to view a smaller region of the graph. It is also a necessary first step in moving from one zoomed in region to another. The user must first reset the graph before zooming in on another region.

4.3.3.3 Average Parameter Values and Metric Scores

The third button on the *Legend* (Figure 4-6) is labeled "Averages". Clicking it displays the average parameter values and metric scores of the selected region (Figure 4-9). The average parameter values are listed on the left side of the window, and the average metric scores are listed on the right side. The average probability of resource availability is also listed on the right side. The Close Window button is at the bottom of the window. Clicking "Averages" on the *Legend* when no sub-region is selected provides the average values for all of the points on the graph. Average values can be viewed for multiple regions of the graph simultaneously by repeatedly selecting regions and clicking "Averages."

The averages display (Figure 4-9) is useful in determining the overall character of a cluster or region of the graph, in terms of the metric scores and parameter values. Although, the average scores for the four plotted metrics can be estimated visually by noting the average location and dot size and color, the other metrics are not visible. The averages display indicates the exact averages of the four metrics plotted for all off the plans within the specified region. In addition, it provides the averages of the remaining three metrics, all of the parameters, and the probabilities. This display gives the user extra information that the graph cannot provide. It also lends itself to quick comparisons of different regions of the graph by viewing the corresponding sets



Parameter	AverageValue	Metric	Average Score
C5-DOV	11.968	Number of Aircraft	48.548
C17-CHS	14.968	System Ute Rate	0.384
C17-RMS	3.677	Operating Cost	7.933E7
C141-WRI	3.839	AMC Hold Time	0.917
C130-RMS	14.903	Flying Hours	7089.396
Port Hold Time	1	TWCF Ute Rate	1.086
Fractioning	0.142	Number of Missions	377.323
Start Frequency	3	Probability	0.001
Flight Sequence	HS		

Figure 4-9: Display of average parameter and metric values for selected region of Figure 4-7

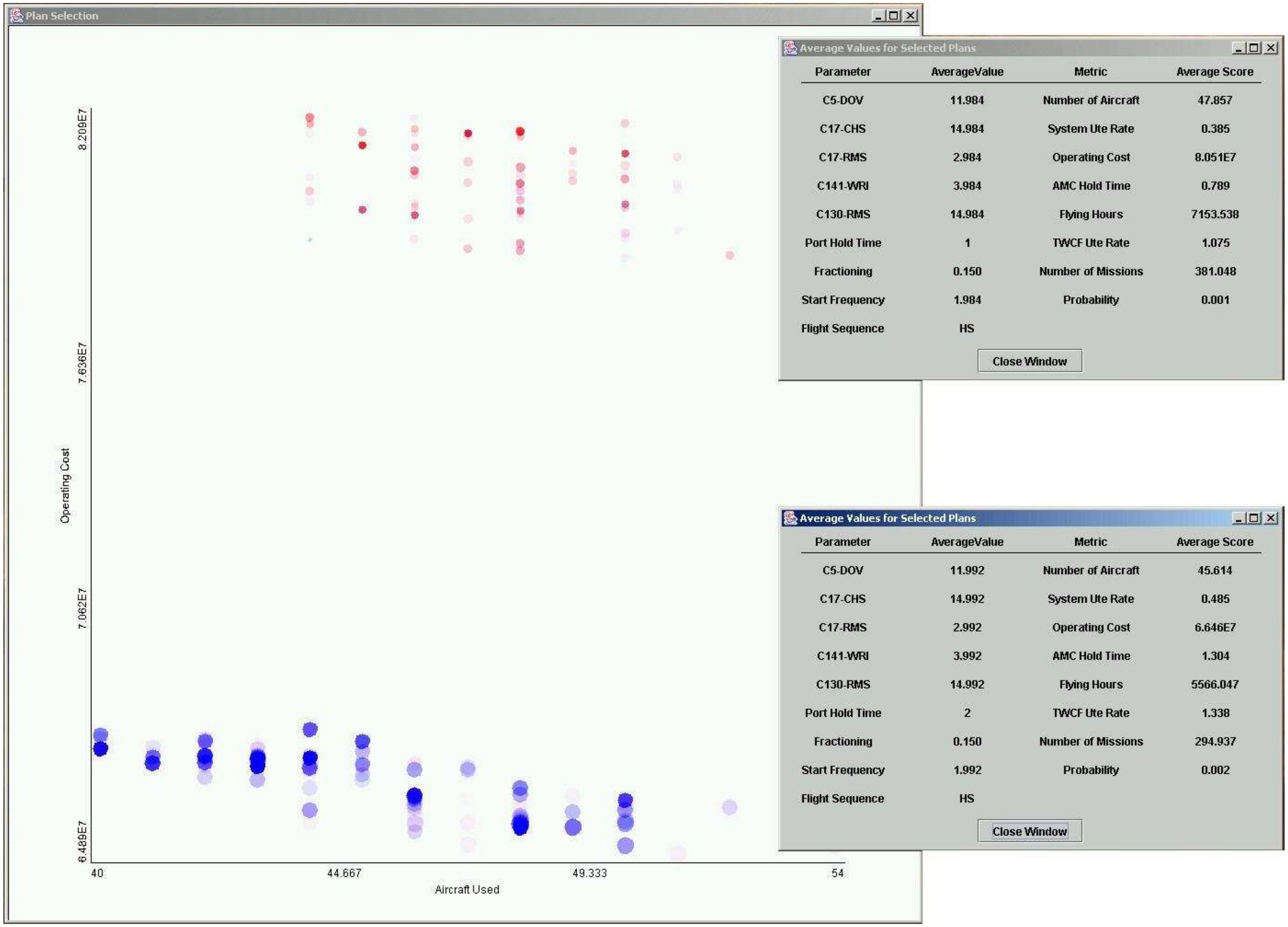
of average values (Figure 4-10). This helps the user identify what distinguishes one region from another.

4.3.4 Pop up Menu

When the user right clicks on a dot in the five dimensional graph, a *Pop Up Menu* containing two submenus appears. The first submenu offers identifying information for the plan (Figure 4-11). It shows the user the values used as inputs to the optimization. The only exception is the probability, which is calculated separately (Section 4.2). The items in this submenu are gray because they are not selectable; they are presented informational for purposes only.

The second submenu contains several options that provide details about the output plan itself (Figure 4-12). All items in this submenu are selectable, and choosing one of them causes a separate data window to be opened. These will be discussed in detail in the section on single plan information (Section 4.4). Table 4.2 points to descriptions and figures of the displays accessed by each option.

The *Pop Up Menu* is a tool for a single plan that is accessible from the *Five-*



Parameter	AverageValue	Metric	Average Score
C5-DOV	11.984	Number of Aircraft	47.857
C17-CHS	14.984	System Ute Rate	0.385
C17-RMS	2.984	Operating Cost	8.051E7
C141-WRI	3.984	AMC Hold Time	0.789
C130-RMS	14.984	Flying Hours	7153.538
Port Hold Time	1	TWCF Ute Rate	1.075
Fractioning	0.150	Number of Missions	381.048
Start Frequency	1.984	Probability	0.001
Flight Sequence	HS		

Parameter	AverageValue	Metric	Average Score
C5-DOV	11.992	Number of Aircraft	45.614
C17-CHS	14.992	System Ute Rate	0.485
C17-RMS	2.992	Operating Cost	6.646E7
C141-WRI	3.992	AMC Hold Time	1.304
C130-RMS	14.992	Flying Hours	5566.047
Port Hold Time	2	TWCF Ute Rate	1.338
Fractioning	0.150	Number of Missions	294.937
Start Frequency	1.992	Probability	0.002
Flight Sequence	HS		

Figure 4-10: Two average parameter and metric value displays. The top display gives information about the upper cluster of red dots, while the bottom one gives information about the lower cluster of blue dots.

Figure 4-11: *Pop Up Menu* showing identifying information (The elements are gray because they are not selectable.)

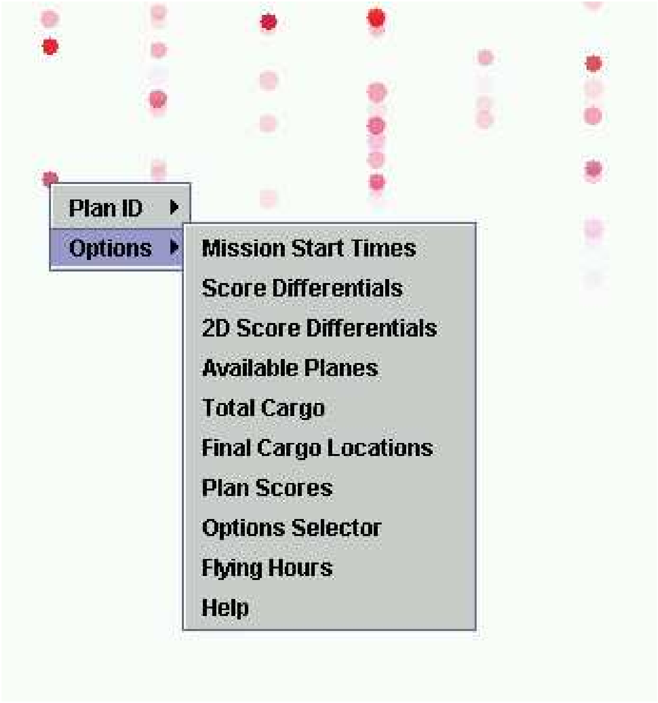


Figure 4-12: *Pop Up Menu* offering plan details

Option	**Describing Section**	**Figure**
Mission Start Times	4.4.2	4-15
Score Differentials	4.4.4.1	4-22
2D Score Differentials	4.4.4.2	4-24
Available Planes	4.4.5	4-25
Total Cargo	4.4.6	4-26
Final Cargo Locations	4.4.6	4-27
Plan Scores	4.4.7	4-28
Options Selector	4.4.1	4-14
Flying Hours	4.4.8	4-29
Help	4.4.9	4-30

Table 4.2: Listing of information accessible from *Pop Up Menu*

Dimensional Graph showing multiple plans. It allows the user to quickly identify and access information about specific plans on the graph. The *Pop Up Menus* can be used in two ways, corresponding to its two submenus. The first submenu (Figure 4-11) can be used to identify a plan on the graph. This will help the user make some quick decisions about whether to continue considering the plan. He could decide that the plan has too many or too few planes available, or that the plan allows too large an AMC Hold Time. He could also decide that the probability of having that set of resources available is too low. The identification submenu gives the user the ability to make judgments on each of the ten input values listed there. He can also sequentially view the identification submenus of multiple plans and make comparisons between them. The second submenu (Figure 4-12) is useful for accessing detailed information about that plan. With a single click, the user can look at detailed information about a specific plan. In later sections, we show how this info is used. The ability to access the information directly from the graph makes it easier for the user to keep track of what he is seeing in the graph.

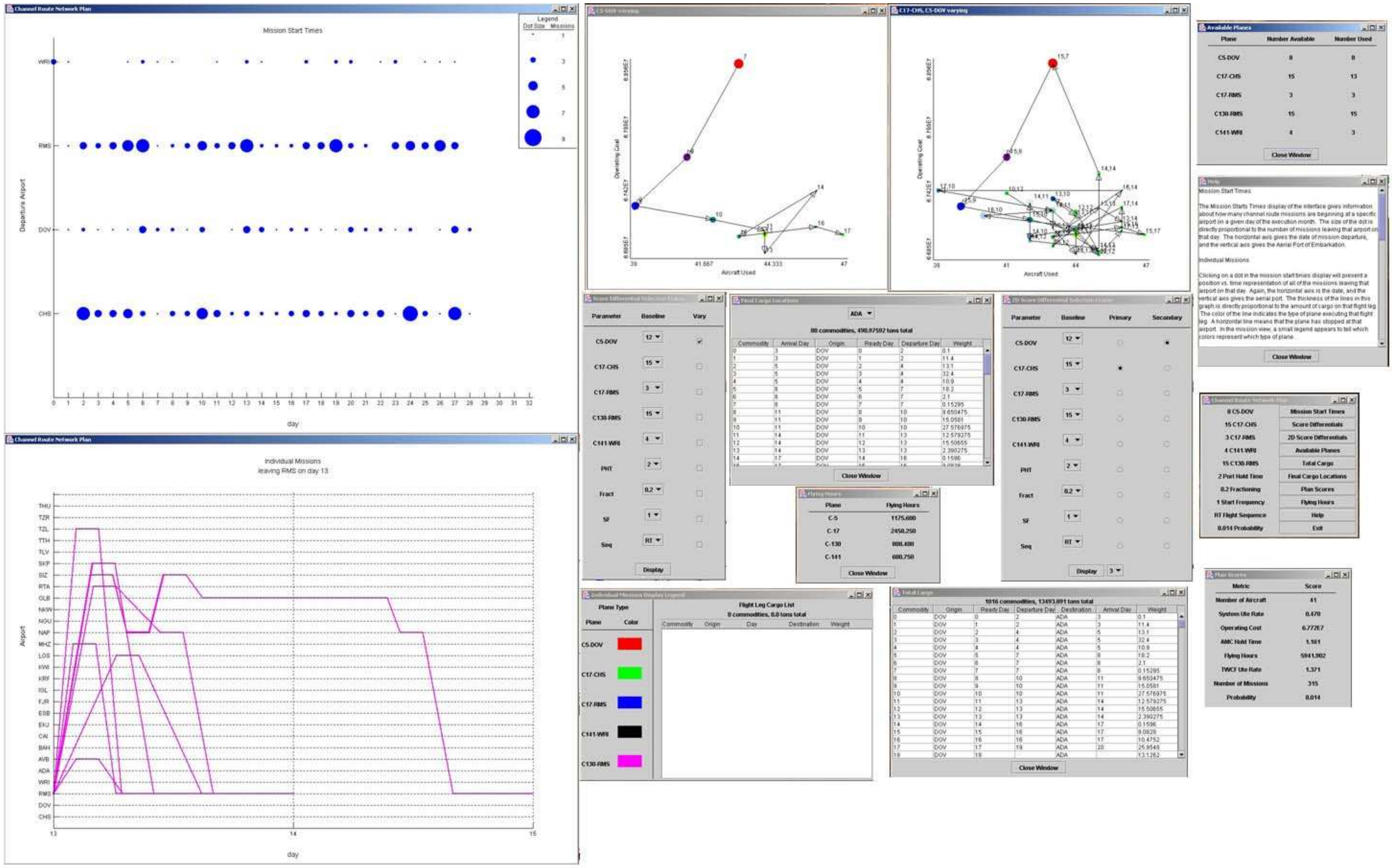
4.4 Viewing a Single Plan

After comparing multiple plans and viewing their general characteristics, the user may want more specific information about a certain plan. ChRIS provides the user with tools reveal the details of a single plan (Figure 4-13). The main component of this tool set is the display of *Mission Start Times*, which leads to information about *Individual Missions*. There are also displays of *Score Differentials*, *Available Planes*, *Total Cargo*, *Final Cargo Locations*, *Plan Scores*, *Flying Hours*, and *Help* information. All of these can be accessed from the *Pop Up Menu* (Section 4.3.4), or from the *Options Selector* (Section 4.4.1).

4.4.1 Options Selector

The *Options Selector* (Figure 4-14) is a centralized point of access to all of the displays of information about a single plan. The inputs used to create that plan are listed on the left side of the selector along with the probability of having that particular set of inputs available. The buttons on the right side of the selector provide access to the various pieces of information about that plan. Table 4.3 lists each of the buttons as well as This tool is important, because it provides the user with an easy way to access all of the information about the plan and reminds them of all the information available. It serves as an anchoring or starting point for their search for that information.

The *Options Selector* serves as an anchoring point for all of the information about a single plan in that the user can access all of the information about a single plan from this window. It is an alternative to using the *Pop Up Menu* from the *Five-Dimensional Graph* points, and it allows the user to access plan information without repeatedly going back to the original *Five-Dimensional Graph* and finding the plan again. For example, the user could be comparing two plans from the graph. Instead of repeatedly accessing the pop up menus for both of the plans, he could bring up the *Options Selector* for both. It is easy to determine which window corresponds to which plan, because the identifying information is listed on the left side of the window. Having both of these windows open allows the user to easily and quickly get information about both plans. From there, he can use all of the individual displays to compare the plans.



Plane	Color
C5-DOV	🟥
C17-CHS	🟩
C17-RMS	🟦
C141-WRI	⬛
C130-RMS	🟪

Parameter	Baseline	Vary
C5-DOV	12 ▾	☑
C17-CHS	15 ▾	☐
C17-RMS	3 ▾	☐
C130-RMS	15 ▾	☐
C141-WRI	4 ▾	☐
PHT	2 ▾	☐
Fract	0.2 ▾	☐
SF	1 ▾	☐
Seq	RT ▾	☐

Display

ADA ▾

80 commodities, 490.87592 tons total

Commodity	Arrival Day	Origin	Ready Day	Departure Day	Weight
0	3	DOV	0	2	0.1
1	3	DOV	1	2	11.4
2	5	DOV	2	4	13.1
3	5	DOV	3	4	32.4
4	5	DOV	4	4	10.9
5	8	DOV	5	7	18.2
6	8	DOV	6	7	2.1
7	8	DOV	7	7	0.15295
8	11	DOV	8	10	9.650475
9	11	DOV	9	10	15.0581
10	11	DOV	10	10	27.576975
11	14	DOV	11	13	12.579275
12	14	DOV	12	13	15.50655
13	14	DOV	13	13	2.390275
14	17	DOV	14	16	0.1596

Close Window

Plane	Flying Hours
C-5	1175.600
C-17	2458.250
C-130	808.400
C-141	600.750

Close Window

Parameter	Baseline	Primary	Secondary
C5-DOV	12 ▾	○	●
C17-CHS	15 ▾	●	○
C17-RMS	3 ▾	○	○
C130-RMS	15 ▾	○	○
C141-WRI	4 ▾	○	○
PHT	2 ▾	○	○
Fract	0.2 ▾	○	○
SF	1 ▾	○	○
Seq	RT ▾	○	○

Display 3 ▾

1016 commodities, 13493.891 tons total

Commodity	Origin	Ready Day	Departure Day	Destination	Arrival Day	Weight
0	DOV	0	2	ADA	3	0.1
1	DOV	1	2	ADA	3	11.4
2	DOV	2	4	ADA	5	13.1
3	DOV	3	4	ADA	5	32.4
4	DOV	4	4	ADA	5	10.9
5	DOV	5	7	ADA	8	18.2
6	DOV	6	7	ADA	8	2.1
7	DOV	7	7	ADA	8	0.15295
8	DOV	8	10	ADA	11	9.650475
9	DOV	9	10	ADA	11	15.0581
10	DOV	10	10	ADA	11	27.576975
11	DOV	11	13	ADA	14	12.579275
12	DOV	12	13	ADA	14	15.50655
13	DOV	13	13	ADA	14	2.390275
14	DOV	14	16	ADA	17	0.1596
15	DOV	15	16	ADA	17	9.0828
16	DOV	16	16	ADA	17	10.4752
17	DOV	17	19	ADA	20	25.9548
18	DOV	18		ADA		13.1262

Close Window

Plane	Number Available	Number Used
C5-DOV	8	8
C17-CHS	15	13
C17-RMS	3	3
C130-RMS	15	15
C141-WRI	4	3

Close Window

Mission Start Times

The Mission Starts Times display of the interface gives information about how many channel route missions are beginning at a specific airport on a given day of the execution month. The size of the dot is directly proportional to the number of missions leaving that airport on that day. The horizontal axis gives the date of mission departure, and the vertical axis gives the Aerial Port of Embarkation.

Individual Missions

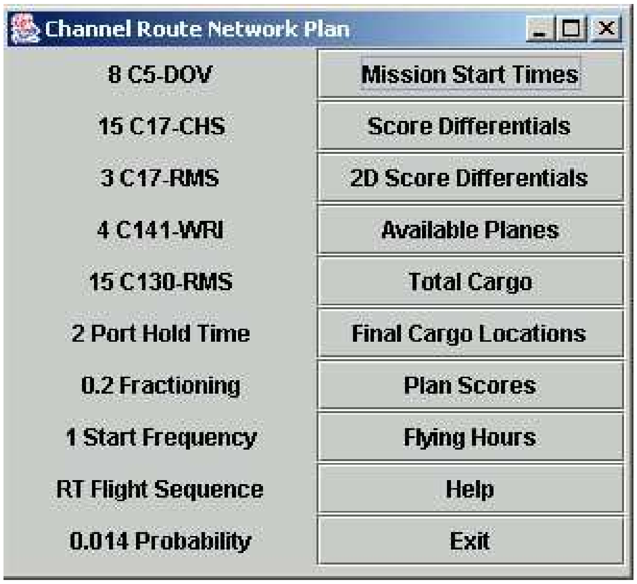
Clicking on a dot in the mission start times display will present a position vs. time representation of all of the missions leaving that airport on that day. Again, the horizontal axis is the date, and the vertical axis gives the aerial port. The thickness of the lines in this graph is directly proportional to the amount of cargo on that flight leg. The color of the line indicates the type of plane executing that flight leg. A horizontal line means that the plane has stopped at that airport. In the mission view, a small legend appears to tell which colors represent which type of plane.

Close Window

8 C5-DOV	Mission Start Times
15 C17-CHS	Score Differentials
3 C17-RMS	2D Score Differentials
4 C141-WRI	Available Planes
15 C130-RMS	Total Cargo
2 Port Hold Time	Final Cargo Locations
0.2 Fractioning	Plan Scores
1 Start Frequency	Flying Hours
RT Flight Sequence	Help
0.014 Probability	Exit

Metric	Score
Number of Aircraft	41
System Ute Rate	0.470
Operating Cost	6.772E7
AMC Hold Time	1.161
Flying Hours	5941.902
TWCF Ute Rate	1.371
Number of Missions	315
Probability	0.014

Figure 4-13: Single plan visualization tool set. The large window in the upper left corner is the display of *mission start times*. The window below it is the display of individual missions. To the right of the *mission start times* graph are two graphs of score differentials, and the tools used to create them are underneath the graphs. There are also displays with information about cargo, planes, flying hours, and metric scores. These are all accesible from the *Options Selector*, which is in the bottom right corner above the display of metric scores.



8 C5-DOV
15 C17-CHS
3 C17-RMS
4 C141-WRI
15 C130-RMS
2 Port Hold Time
0.2 Fractioning
1 Start Frequency
RT Flight Sequence
0.014 Probability

Figure 4-14: *Options Selector* facilitating the selection of single plan information

Button	**Describing Section**	**Figure**
Mission Start Times	4.4.2	4-15
Score Differentials	4.4.4.1	4-22
2D Score Differentials	4.4.4.2	4-24
Available Planes	4.4.5	4-25
Total Cargo	4.4.6	4-26
Final Cargo Locations	4.4.6	4-27
Plan Scores	4.4.7	4-28
Flying Hours	4.4.8	4-29
Help	4.4.9	4-30

Table 4.3: Listing of information accessible from the *Options Selector*

4.4.2 Mission Start Times

When viewing the details of a specific plan, the primary information shown is the display of *mission start times* (Figure 4-15). Recall that a “mission” is a set of flight legs between multiple airports. The plane flying the mission takes off from and lands at each of the airports, potentially gaining or losing cargo at each one. This display gives information about how many channel route missions are beginning at a specific airport on a given day of the month. The diameter of the dot is directly proportional to the number of missions leaving that airport on that day. There is a small legend in the upper right hand corner that translates dot size to a number of missions. The horizontal axis gives the date of mission departure, and the vertical axis gives the departure airport. The user can also click on a dot to access information about the missions leaving that airport on that day (Section 4.4.3).

The *Mission Start Times* graph (Figure 4-15) tells the user how many missions are leaving an airport on a given day. This information is useful in two ways. First, it tells the user which of the departure airports are heavily tasked. For example, in Figure 4-15, many more missions are leaving CHS and RMS than DOV and WRI. In fact, on most days of the month, at most one mission is leaving WRI, while CHS routinely has 8 missions leaving on a given day.

The second kind of information conveyed is the repetitive nature of the plans generated by the optimization. There is a tendency for the plans to take a weekly structure, where similar sets of missions occur once a week on the same day of the week. For example, from Figure 4-15, on day six, there are seven missions leaving RMS, and again on day 13, there are seven missions leaving RMS. Clicking on each of these dots reveals that the sets of seven missions for the two days are very similar.

Aside from its use as a data display, this graph serves as a gateway to the display of mission details. Single clicking on a dot in the graph brings up the detailed display of *Individual Missions* (Figure 4-16). If the user wants to view information about all the missions leaving CHS on day 15 of the month, he can access it simply by clicking on the appropriate dot.

This graph is important because it provides an overview of the plan for the entire month with the option of accessing more detailed information. This level of presentation prevents overloading the user with information about specific missions. This display allows the user to view only those missions in which he is interested. Isolating those missions will allow him to better understand the details of that mission.

4.4.3 Individual Missions

Clicking on a dot in the display of *mission start times* (Figure 4-15)will present a position vs. time representation of all of the missions leaving that airport on that day (Figure 4-16). It is very similar to the Tufte's train schedule graph [13]. The horizontal axis is time, divided by days, and the vertical axis gives the airport. The thickness of the lines in this graph is directly proportional to the amount of cargo on that flight leg. A horizontal line means that the plane has stopped at that airport. The color of the line indicates the type of plane executing that flight leg. For example, the top mission in the figure leaves CHS and makes stop CHS and makes stops in DOV and TLV before returning to CHS. The bottom mission makes stops in DOV and CAI between leaving from and returning to CHS.

In this individual mission view, a legend that tells the user two things. The left side tells which colors on the display represent which type of plane (Figure 4-17). The right side displays the cargo carried on a particular flight leg of a mission. When the user clicks on the line representing a flight leg, the cargo list for that plane for that flight leg is shown in the legend. The total number of cargo commodities and the total weight is also listed above the text field.

Planes often leave their base airport with no cargo, and they are filled with cargo at one of their stops. This is evident in Figure 4-18, which is a zoomed in region of Figure 4-16. The planes left their base airport of CHS for NGU carrying no cargo, and the line connecting CHS to NGU is thin. At NGU, the planes took on cargo, represented by the thicker lines. After leaving NGU, one plane drops its cargo off at NAP, and that line becomes thin again at that point. Likewise, another plane departs NGU for SIZ, and leaves cargo there. These changes in tonnage carried are reflected

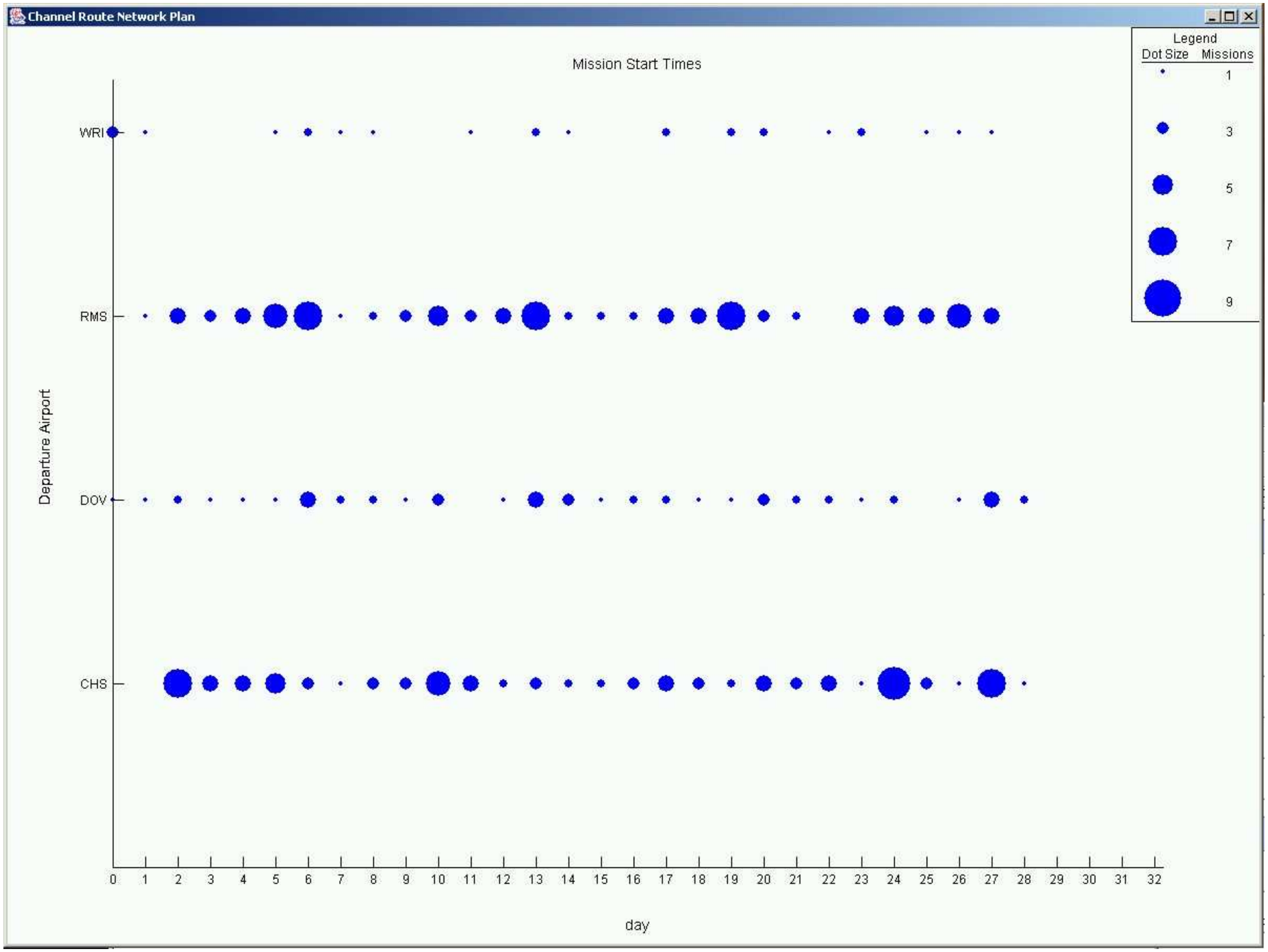
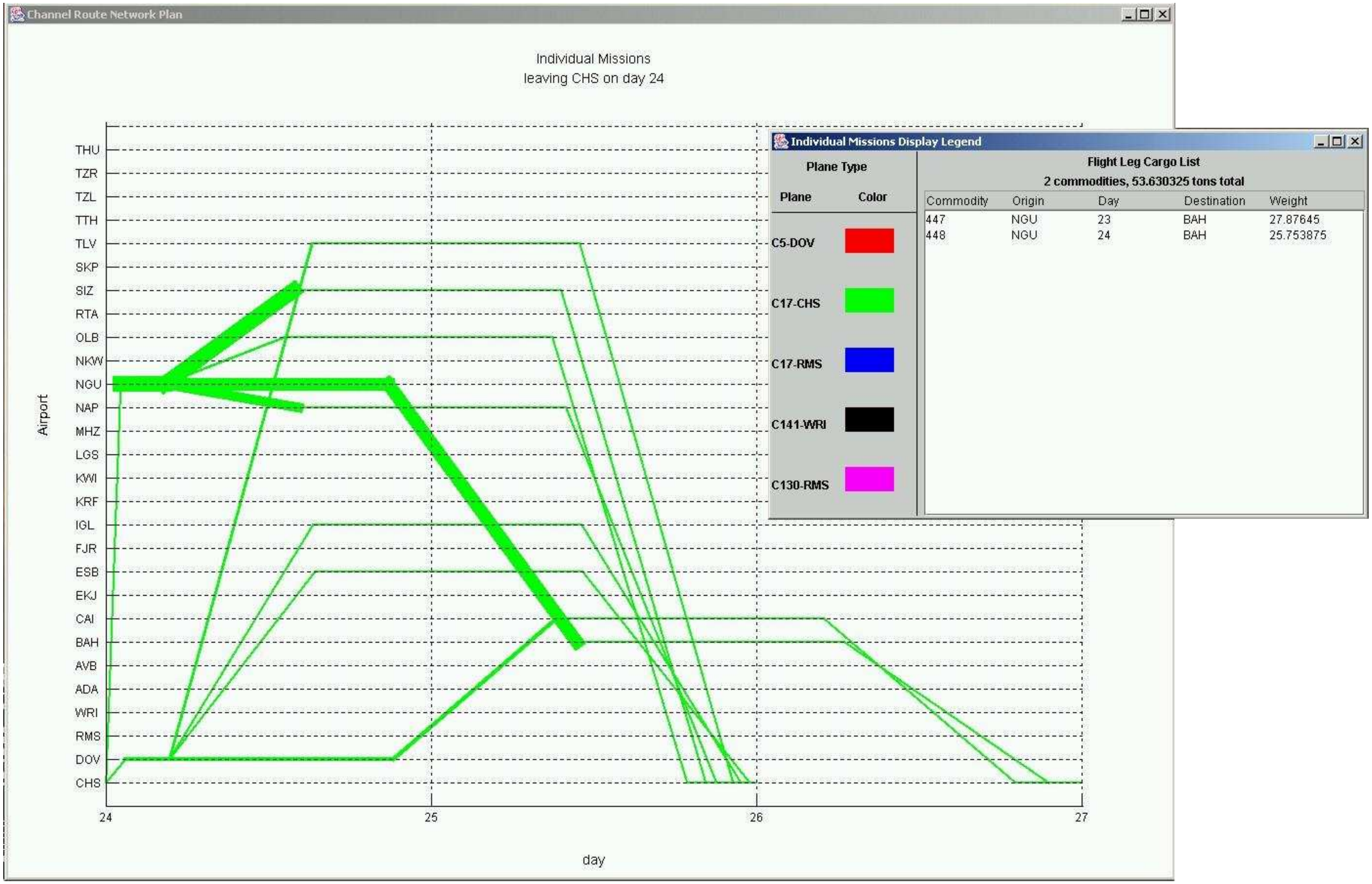


Figure 4-15: *Mission Start Times* tool. Dot size indicates the number of missions beginning at the airport on the day



Plane Type

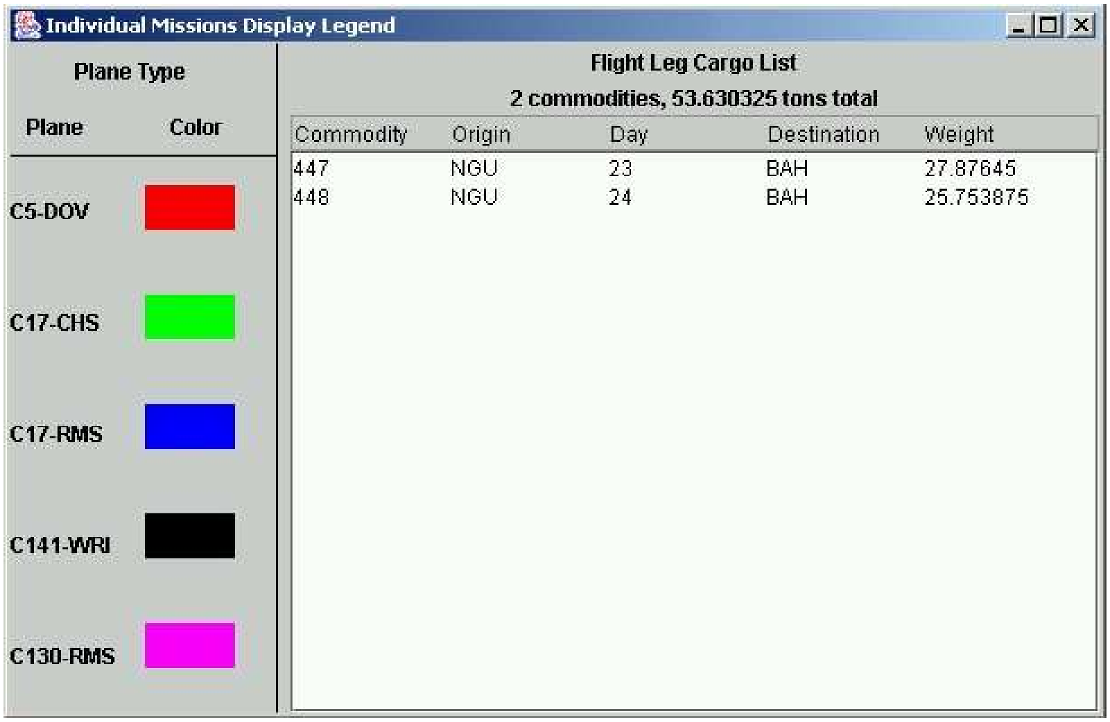
Plane	Color
C5-DOV	🟥
C17-CHS	🟩
C17-RMS	🟦
C141-WRI	⬛
C130-RMS	🟪

Flight Leg Cargo List

2 commodities, 53.630325 tons total

Commodity	Origin	Day	Destination	Weight
447	NGU	23	BAH	27.87645
448	NGU	24	BAH	25.753875

Figure 4-16: Visualization of missions leaving CHS on day 24 using C17 planes. Thickness of lines indicates cargo quantity on a flight leg and color indicates the type of plane flying the mission.



Plane	Color
C5-DOV	🟥
C17-CHS	🟩
C17-RMS	🟦
C141-WRI	⬛
C130-RMS	🟪

Flight Leg Cargo List

2 commodities, 53.630325 tons total

Commodity	Origin	Day	Destination	Weight
447	NGU	23	BAH	27.87645
448	NGU	24	BAH	25.753875

Figure 4-17: Legend for *Individual Missions* display

in the thickness of the lines throughout the map of flights.

One artifact of this presentation of mission flight paths is that when multiple planes are flying between two bases at the same time, only one line is visible. It only becomes clear that there were multiple planes flying that route when they branch out for different airports. This is shown in close up in Figure 4-19, where four planes left CHS for DOV, but only one line is visible. It is clear that there were four planes making that flight when this single line splits into four in the middle of the figure.

Displays of *Individual Missions* like the one given in Figure 4-16 represent the most detailed graphical information one can get about the actual flights in the plan. They show the flight legs that make up the missions leaving an airport on a given day, as well as the cargo being carried on those flight legs. These displays are informational in that nothing can be inferred from what they show. They simply present the structure and the details of the missions being flown. However, they are still useful for verifying how cargo is flowing through the network. He can simply determine the actual flights in each mission by following the connected line segments through the graph. The

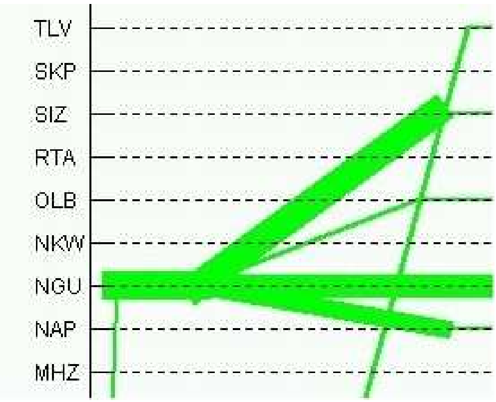


Figure 4-18: Detail of Figure 4-16 showing planes adding significant amounts of cargo at NGU

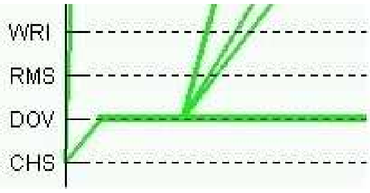


Figure 4-19: Detail of Figure 4-16 illustrating that the displayed lines for four flights are overlapped from CHS to DOV and then divide from there to separate destinations.

graphs also allow the user to visually determine the overall structure of the missions displayed. This allows the user to quickly compare displays of missions for different day/airport pairs. By viewing multiple displays of *Individual Missions* at once, the user can compare how the missions for each day/airport pair are shipping their cargo. Clicking on the successive flight legs and using the legend's cargo list makes it possible to trace the path of cargo through the network. By clicking on successive flight legs in a mission, the user can determine when cargo is loaded onto a plane and when it is taken off. In addition, the labels on the x-axis tell the user how long the missions are taking.

4.4.4 Score Differentials

To evaluate a plan, the user may want to identify how a plan's metrics vary as its inputs are perturbed. One method for doing this is with traditional shadow prices from optimization theory (Section 2.1). However, we determined that shadow prices are insufficient because they are not guaranteed to be valid over the entire range between two plans. Instead, *score differentials* are used. Score differentials are the discrete differences in metric scores for two plans. There are two displays that use score differentials. One uses score differentials to show how metric scores change as one input varies, while the second uses them to show how metric scores change as two parameters vary simultaneously.

4.4.4.1 One-Dimensional Score Differentials

The goal of the *One-Dimensional Score Differentials* tool is to show how individual changes in the inputs cause changes in the values of the various metrics displayed on the five dimensional graph. For example, the graph in Figure 4-20 shows the user how the various metric scores change when 8 C5-DOV planes are available instead of 7. Aircraft Used (x), Operating Cost (y), Missions Flown (color), and TWCF Ute Rate (size) are plotted on the graph. In this case, Aircraft Used metric decreases, the Operating Cost decreases, the number of Missions Flown decreases, and the TWCF Ute Rate decreases. Simple differences in metric scores are used to tell how the score changes when the inputs are perturbed from a baseline. We will call these differences in metric scores *score differentials*. See Section 2.1 for a basis of these score differentials from linear optimization theory.

The user identifies what *One-Dimensional Score Differentials* to display using the tool in Figure 4-21. All of the input parameters to the optimization are listed in the left column. The user chooses values for each of the parameters using the drop-down selection boxes in the middle column of the tool. Each selection box contains the values of that parameter for which runs of the optimization have been done. Collectively, these selections will form a baseline around which the inputs will be perturbed independently. Finally, in the rightmost column are check boxes, which the user can select to indicate that he would like to view a graph with variations in that parameter.

Clicking the Display button on the bottom of the Selection Frame creates a *Five-Dimensional Graph* for each of the input parameters checked to be varied. The title

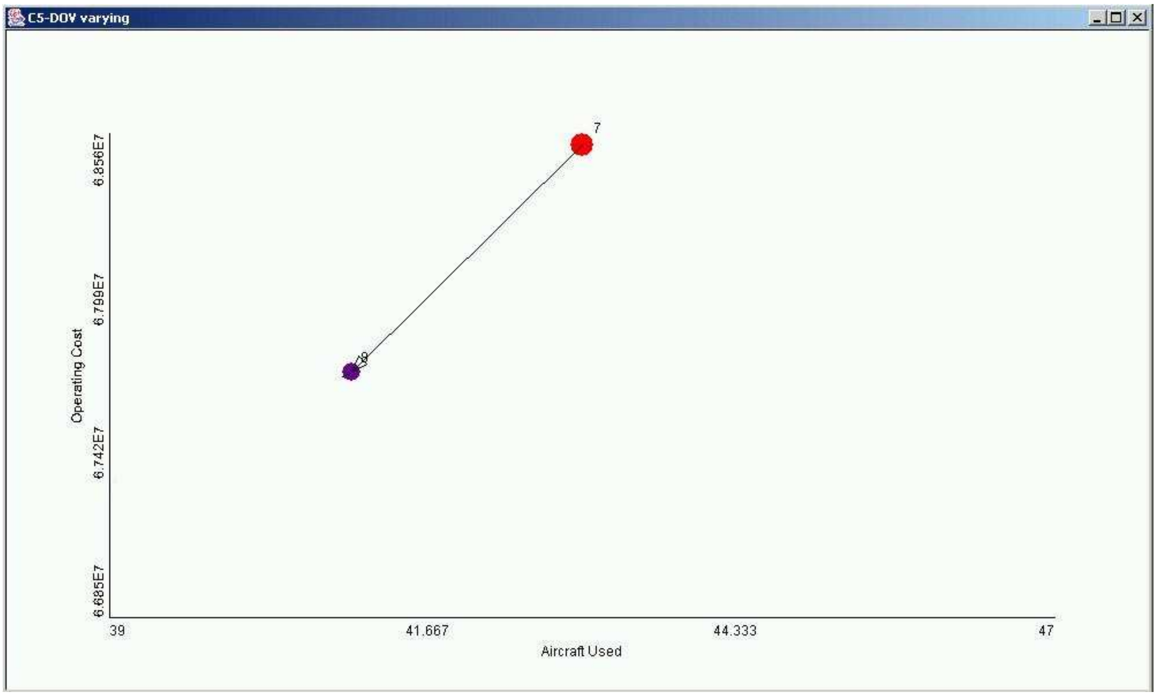
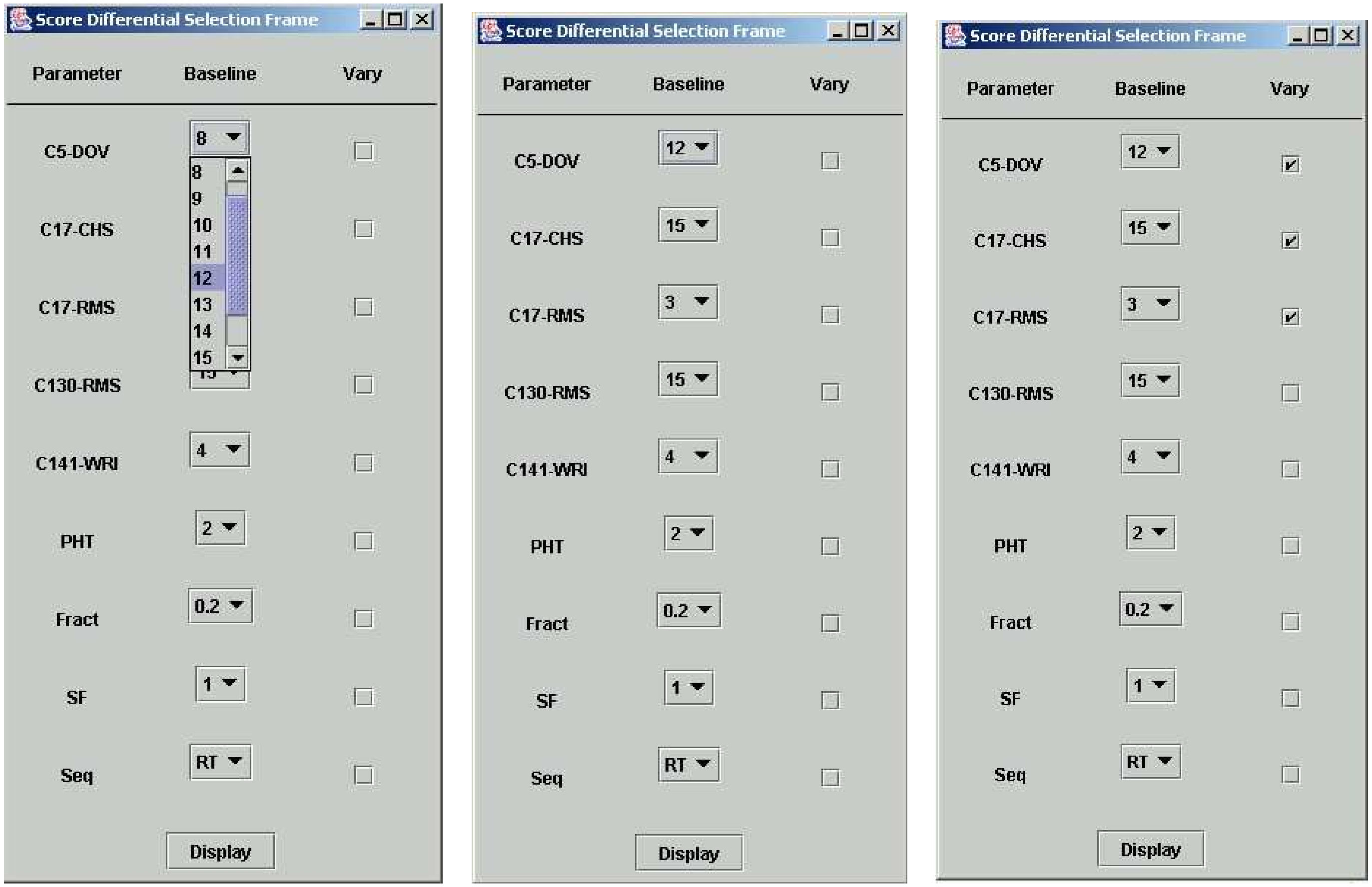


Figure 4-20: Example of Score Differentials, or changes in scores due to changes in input parameter values

of the window containing each graph identifies the parameter being varied. When multiple parameters are checked, these graphs are tiled on the screen so that they are all visible at once (Figure 4-22). The scales are the same for all of the graphs, and a single legend is presented for all of them. Furthermore, if the user draws a rectangle on one graph, the system draws the same rectangle on all of the other graphs. Zooming in on one graph also zooms in on all of the other graphs, and resetting one graph resets them all.

The individual *Five-Dimensional Graphs* in Figure 4-22 are very similar to the one in Figure 4-3, with a few important changes. The graphs in Figure 4-22 do not display all plans. Instead, it displays those plans that differ from the baseline plan in the parameter being varied. For example, the number of available C5-DOV planes is varying in the top left graph of Figure 4-22. The only those plans that are displayed on this graph are those different from the selected baseline plan in the number of available C5-DOV planes. The baseline plan is indicated by a yellow box in each graph.



(a)

Parameter	Baseline	Vary
C5-DOV	8 ▼ 8 9 10 11 12 13 14 15	☐
C17-CHS		☐
C17-RMS		☐
C130-RMS		☐
C141-WRI	4 ▼	☐
PHT	2 ▼	☐
Fract	0.2 ▼	☐
SF	1 ▼	☐
Seq	RT ▼	☐

(b)

Parameter	Baseline	Vary
C5-DOV	12 ▼	☐
C17-CHS	15 ▼	☐
C17-RMS	3 ▼	☐
C130-RMS	15 ▼	☐
C141-WRI	4 ▼	☐
PHT	2 ▼	☐
Fract	0.2 ▼	☐
SF	1 ▼	☐
Seq	RT ▼	☐

(c)

Parameter	Baseline	Vary
C5-DOV	12 ▼	☑
C17-CHS	15 ▼	☑
C17-RMS	3 ▼	☑
C130-RMS	15 ▼	☐
C141-WRI	4 ▼	☐
PHT	2 ▼	☐
Fract	0.2 ▼	☐
SF	1 ▼	☐
Seq	RT ▼	☐

Figure 4-21: Use of *One-Dimensional Score Differentials* Selection Tool for selecting which parameters to vary around which baseline for the *One-Dimensional Score Differentials* display. (a) shows baseline selection, (b) shows the baseline fully selected, and (c) shows selection of which parameters to vary.

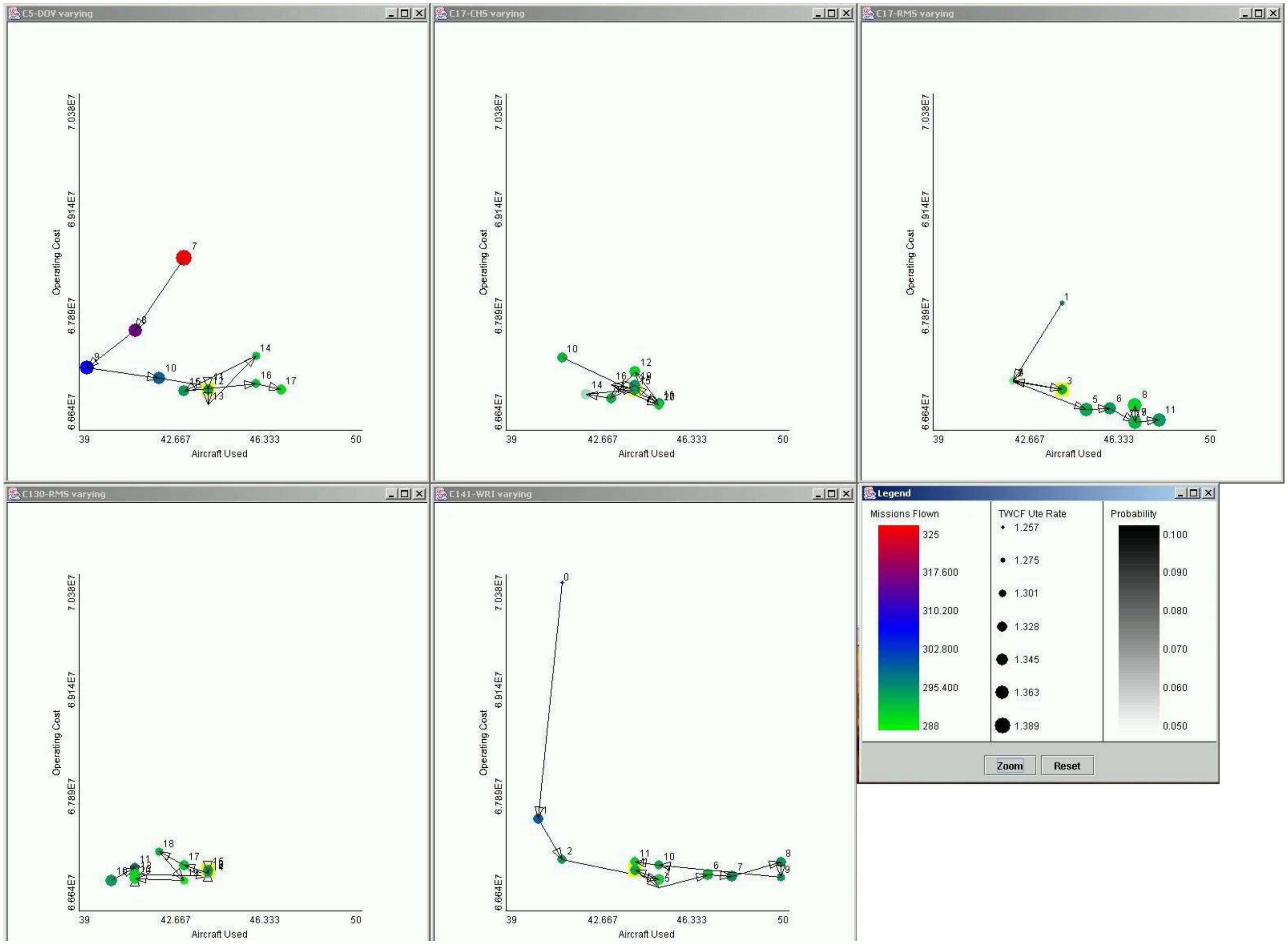


Figure 4-22: *One-Dimensional Score Differentials*: Five independent graphs showing variations of a single parameter around a baseline plan. The tile of each window identifies the varied parameter.

In addition, some extra information is added to the basic *Five-Dimensional Graph*. Each point is labeled with its value for the varying parameter. The points on the graph are connected by arrows; an arrow from plan A to plan B means that plan B has a larger value for the parameter being varied. The arrows connect the plans in order of smallest to largest values of the varying parameter. For example, in the top left graph of Figure 4-22, there is an arrow going from the point labeled with a 7 to the point labeled with an 8, since 8 is the next highest amount of C5-DOV planes available after 7. However, the step size between each pair of plans connected by an arrow may vary, based upon what plans are available for display.

When viewing these graphs, the user can follow the arrows and the labels to determine how the varying parameters are changing. Based upon the changes in location, and dot color and size, the user can qualitatively determine how the plotted metric scores are changing. For example, in the top right graph of Figure 4-22, there is not much change in the dot size or color, except for the point labeled 1 is smaller than the others. This means that the Missions Flown and TWCF Ute Rate do not vary greatly when the number of C17-RMS varies. All of the points are also clustered closely on graph, indicating that the Aircraft Used and Operating Cost are not changing much between plans. When the user selects or zooms in on an area of one graph, it indicates that he is only interested in plans in that area. Therefore, the same area is selected or zoomed in on each of the other graphs, so that the user is not distracted by things in which he is not interested. If the user were to zoom in on and select a region of the top middle graph of Figure 4-22 showing variations in C17-CHS, the same selection and zooming would occur in all five graphs.

This information on *One-Dimensional Score Differentials* can be used in three ways. First, it can be used to make a case for a different set of resources: the information presented on a score differential graph may allow the user to say that a set of resources different from the proposed set should be used, because the plan associated with the new set is better. For example, in Figure 4-20 making 8 C5-DOV planes available instead of 7 results in a decreased operating cost, which suggests trying to arrange for that extra plane to save money. The metrics plotted on the score differential graph are the same as those plotted on the original *Five-Dimensional Graph*, so only differences in score for those metrics plotted on the graph can be used in making a case for a different set of resources.

Second, score differential information is useful in doing contingency planning, based on the results of expected changes. We take contingency planning to mean choosing the best plan while being aware of the ramifications of expected changes to the set of inputs used to create that plan. For example, contingency planning would be selecting the best plan, which needs 8 available C5-DOV planes while being aware of what would change if that number changes to 7. In this case, the user would look for distance, size, and color changes on the graph of score differentials, representing the differences in metric scores between the two plans. The user should note how the metric scores will change for expected changes in the input parameters. For example, consider again Figure 4-20. Imagine that the user is going to execute a plan involving 8 C5-DOV planes, but that he expects to lose one. From the graph, he can report that the Operating Cost will go up, and the Aircraft Used will go up as the number of C5-DOV planes available changes from 8 to 7. The plan involving 8 C5-DOV planes can be executed, and the planners will be aware of what will change if one plane is lost.

Third, the information is useful for checking robustness. Remember that we define robustness as independence from external effects, that change which resources are available. In robustness checking, the goal is to minimize the effects of expected changes while planning. This means that the plan will be robust to complete subsitution of one plan for another before execution starts. This is important because there is less time to plan as the execution period nears. Finding a robust plan minimizes the amount of time spent replanning and reviewing the plan space. Instead, when the changes occur, a similar plan that has already been found can be substituted for the original, now impossible, plan.

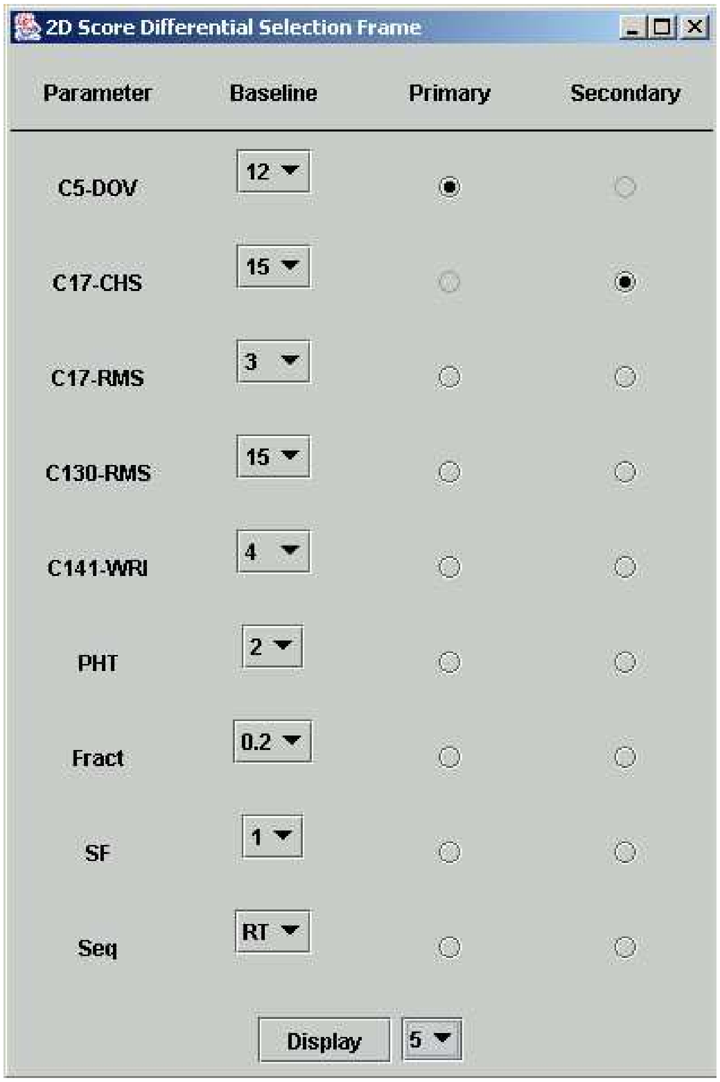
The main thing to look for on the graphs of score differentials is a cluster of dots around the baseline plan, which uses the current set of resources. This indicates that changes in the parameter that is being varied do not cause large changes in that range of the plotted metric scores. The plan can switch between any of the points in the cluster during the planning process without a large change in the metric scores, which indicates robustness. For example, in Figure 4-22, the top middle graph, which shows variation in the number of C17-CHS planes, and the bottom left graph, which shows variation in the number of C130-RMS planes, exhibit tight clustering of points around the baseline. If such a cluster exists around the baseline plan, which uses the current set of resources, then the plan can be called robust to changes in the varying parameter that occur while planning.

This capability is very important, because variations in the inputs are extremely likely. It is very beneficial to the user to be able to see how the characteristics of the plan change when the set of available resources changes. These graphs give the user the ability to see how changes to a single parameter affect the values for the various metrics on the graphs. The arrows allow the user to quickly identify which way represents an increasing amount of resource. They allow the user to say exactly what changes when the input is varied. Therefore, these graphs quickly give the user much information about the consequence of changes in the amount of available resources.

4.4.4.2 Two-Dimensional Score Differentials

The limitation of the *One-Dimensional Score Differential* tool is that it forces the user to consider variations in only one parameter at a time, or a set of independently varying parameters (Figure 4-22). The two-dimensional score differential tool allows the user to consider two variations simultaneously. The tool for selecting which parameters to vary (Figure 4-23) is similar to the one for *One-Dimensional Score Differentials*. The left column lists the input parameters, and the baseline can be chosen using the drop-down selection boxes in the second column. On the right are two columns of radio buttons, which are used to select the two parameters (primary and secondary) to be varied. Only one button in each column of buttons can be chosen.

In addition to the display button at the bottom of the window, there is another drop-down selection box. This indicates the number of values (n) of the primary



2D Score Differential Selection Frame

Parameter	Baseline	Primary	Secondary
C5-DOV	12	●	○
C17-CHS	15	○	●
C17-RMS	3	○	○
C130-RMS	15	○	○
C141-WRI	4	○	○
PHT	2	○	○
Fract	0.2	○	○
SF	1	○	○
Seq	RT	○	○

Figure 4-23: *Two-Dimensional Score Differentials* Selection Tool used for selecting which two parameters to vary around a baseline

parameter to view. In response to the user's click of the display button, n+1 graphs are created (Figure 4-24). As in the one-dimensional case, the graphs are sized and tiled to all be visible at once, are scaled identically, and manipulating one causes manipulations in the others.

With the exception of the "root" graph shown in the top left corner of Figure 4-24, the other, subordinate, graphs are the same in structure to those shown in Figure 4-22 in that they show variation in a single parameter. The subordinate graphs show variations in the secondary parameter for fixed values of the primary parameter. Each of values for the primary parameter is shown in a separate graph, each of which shows variations in the second varying parameter.

The root graph shows variations in both parameters, essentially combining information from all of the subordinate graphs. It displays all of the plans that differ from the baseline only in the two parameters that are being varied. The ordering of the plans for the purposes of drawing arrows on the graph is first in order of increases in the parameter parameter, then in order of increases of the secondary parameter. All points are labeled with a pari of numbers that are, respectively, the values of the primary and secondary parameter for that plan.

Two-Dimensional Score Differentials (Figure 4-24), where the effects of simultaneous variations in two input parameters are noted, are used for the same purposes as one-dimensional score differentials. When trying to make a case for a different set of resources, the user should check how changes in the two varying parameters affect the plotted metrics. This is most easily done using the root graph from the display of two-dimensional score differentials. This graph shows variations in both parameters at once, allowing the user to see how the metric scores change when both parameters are changes simultaneously. Based on this information, the user may be able to say that a different set of resources should be used, because metric scores of the associated plan are better than those of the current plan.

Contingency planning is also most easily done from the root graph from the display of *Two-Dimensional Score Differentials*. Again, all the user has to report is how much the metric scores will change for expected changes to the inputs from those of the

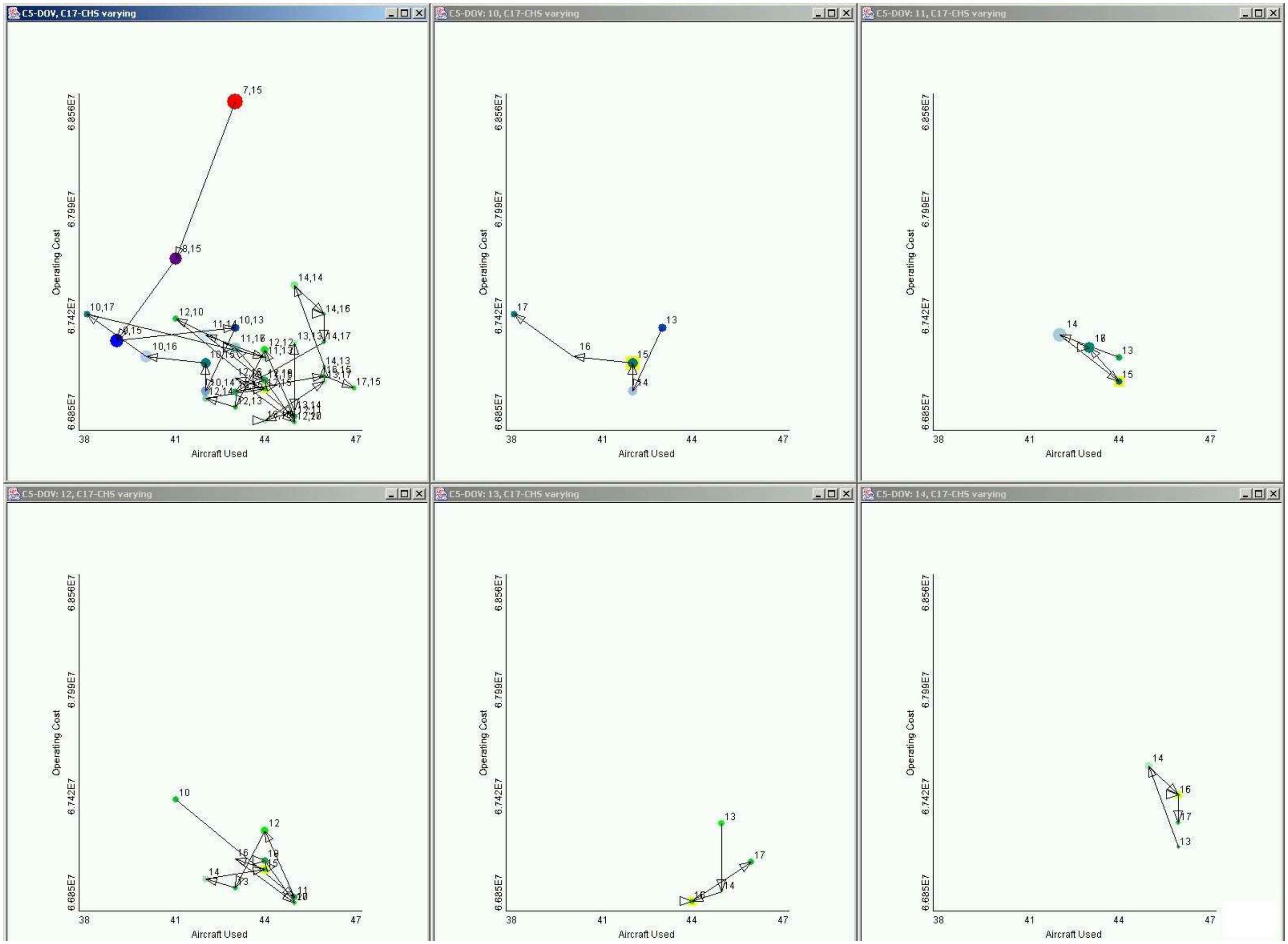


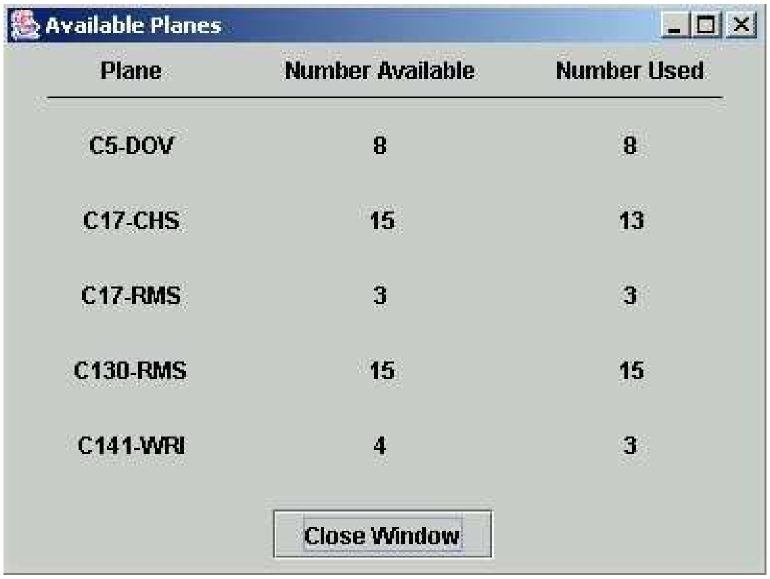
Figure 4-24: *Two-Dimensional Score Differentials*: Six related graphs showing variations in two parameters simultaneously

selected best plan. This can be done by finding the appropriate points on the graph and noticing the distance, size, and color differences between them, as is done for *One-Dimensional Score Differentials*. It can also be done by finding the appropriate points on the other graphs that show single parameter variations. However, variations in two parameters would involve points on two graphs, and information would have to be combined by the user from those graphs.

For example, consider Figure 4-24. Imagine that the user has decided that the best plan uses 13 C5-DOV planes and 13 C17-CHS, and he wants to report the effects of adding 1 C5-DOV plane and 1 C17-CHS plane. He can find both points on the root graph to determine that the cost will go up, while the other three metrics stay the same. He can also make this determination by combining information from the bottom middle (graph A) and bottom right graphs in the display. The bottom middle graph (graph B) shows variations in the number of C17-CHS planes when the number of C5-DOV planes is held at 13. The bottom right is similar, except the number of C5-DOV planes is held at 14. The selected best plan is labeled on graph A with a 13, and the plan representing the expected variation is labeled with a 14 in graph B. From these two graphs, the user can visually combine both graphs to see that the Operating Cost increases and the other metrics stay about the same when moving from the selected point in graph A to the selected point in graph B.

Checking for robustness of a plan using two-dimensional score differentials involves looking at all of the graphs. If clustering exists in all of the graphs, then the original plan can be called robust. This is because all of the possible changes of the two varying parameters result in small changes in the plotted metric scores. Therefore, the plan can switch between any of those plotted without a great change in metric scores. For example, in Figure 4-24, the all but the top middle graph exhibit the clustering that is indicative of robustness. Because there is a graph where tight clustering does not exist, the plan cannot be called robust.

This tool is useful for the similar reasons as the display of the *One-Dimensional Score Differentials*, but it further allows the user to see how simultaneous changes in two parameters affect the scores of the various metrics being displayed. It may be



Plane	Number Available	Number Used
C5-DOV	8	8
C17-CHS	15	13
C17-RMS	3	3
C130-RMS	15	15
C141-WRI	4	3

Figure 4-25: *Available Planes* tool showing available and used planes

difficult to attribute the changes in metric scores to one of the varying parameters when looking at these graphs, but the user can still determine how the pair of changes affects the scores.

4.4.5 Available Planes

Several of the inputs to the optimization are the number of planes of each type that are available. This information can be displayed from either the *Pop Up Menu* (Figure 4-12) or the options selector (Figure 4-14) by clicking on the *Available Planes* option. This creates a report window with information about the planes that were available to fly channel route missions when a particular plan was created (Figure 4-25). For each type of plane, the number available and the number actually used are listed.

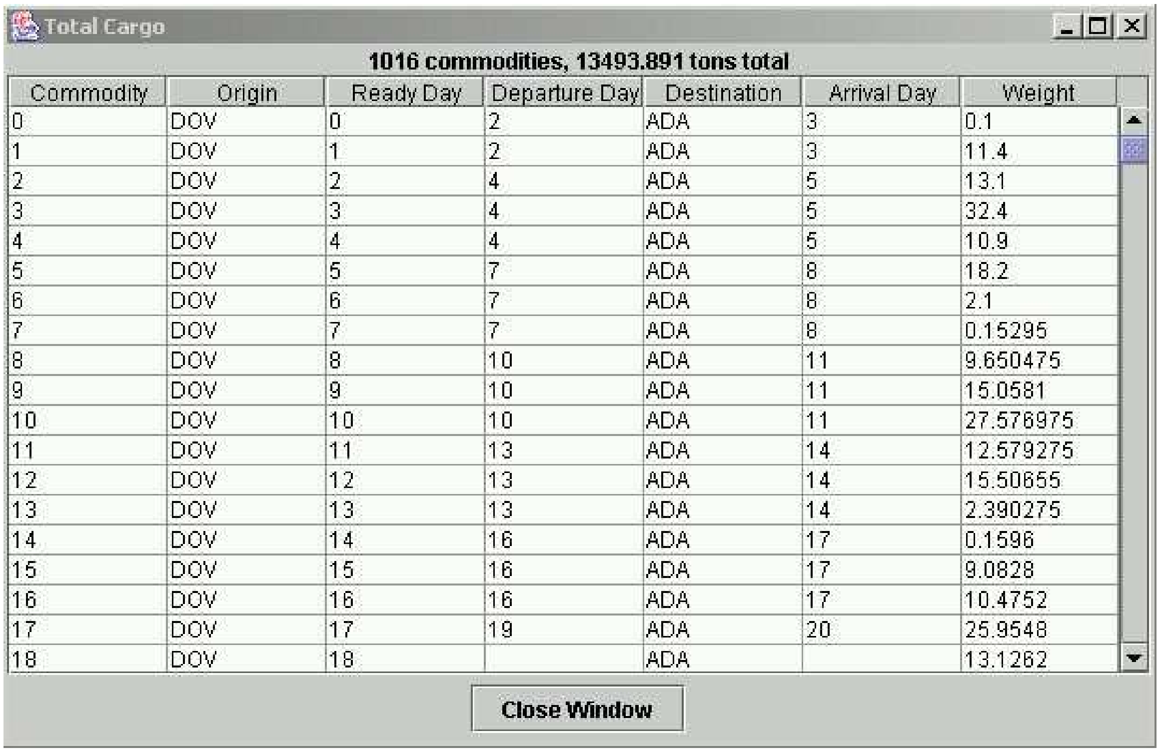
The number of available planes is used as a key piece of identifying information for a plan. Therefore, it is useful in reminding the user which plan he is looking at. It is also useful to see the actual number of planes that are being used and how it compares to the number available. This information indicates whether or not the choice of available resources is a good one. For example, a user may decide that a plan is undesirable if it is not using all planes made available to fly channel route missions, for those unused planes could be used elsewhere.

The information in this display is a more specific version of the Aircraft Used metric because it reports information for individual aircraft types. It can be used in conjunction with the listing of metric scores to answer questions about the number of planes used. For example, consider that the user knows that five of the C5-DOV planes that were said to be available will need maintenance soon and that he would like a plan that does not need to use those planes. Seeing that a certain number of total planes are used is not enough information to determine this. In this case, the *Available Planes* display will tell the user how many C5-DOV planes were available and how many were used in the plan. If the number used is five less than the number available, then the plan is acceptable.

4.4.6 Total Cargo and Final Cargo Locations

The Channel Route Network Planning problem deals with the shipment of cargo, and these two displays provide information about that cargo being shipped. Clicking on the *Total Cargo* option from either the *Pop Up Menu* (Figure 4-12) or the *Options Selector* (Figure 4-14) creates a report window with information about all that cargo being shipped (Figure 4-26). Each cargo commodity is listed by commodity number, along with its origin airport, day it is ready for shipping, day of departure, destination airport, day of arrival, and weight in tons. Clicking on the labels at the top of each column of the list of cargo commodities will sort the list according to that column. The information at the top of the display presents the total number of cargo commodities and their aggregate weight.

Clicking on the *Final Cargo Locations* Button creates a smaller window that shows where each of the cargo commodities ends up (Figure 4-27). The information is organized by airport. Selecting an airport from the drop down list at the top brings up a list of all the cargo commodities that are supposed to be at that airport at the end of the plan. Each commodity is listed by number with its day of arrival, origin airport, day it is ready for shipping, day of departure, and weight in tons. Clicking on the labels at the top of each column will sort the list according to that column.



1016 commodities, 13493.891 tons total

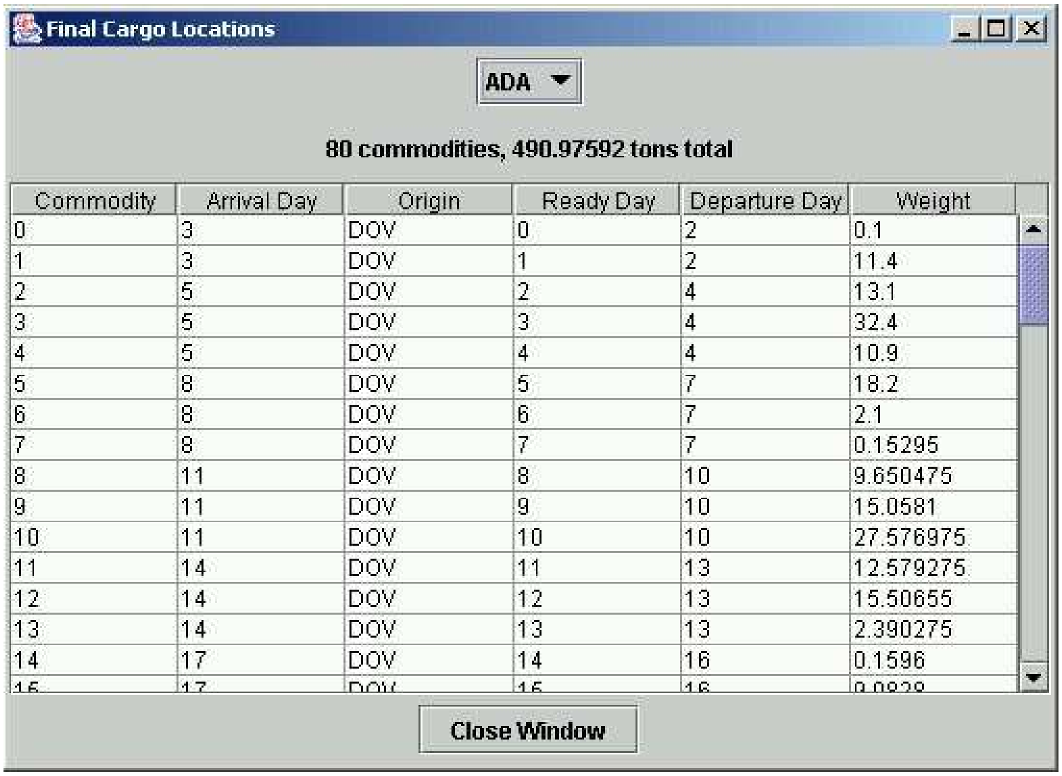
Commodity	Origin	Ready Day	Departure Day	Destination	Arrival Day	Weight
0	DOV	0	2	ADA	3	0.1
1	DOV	1	2	ADA	3	11.4
2	DOV	2	4	ADA	5	13.1
3	DOV	3	4	ADA	5	32.4
4	DOV	4	4	ADA	5	10.9
5	DOV	5	7	ADA	8	18.2
6	DOV	6	7	ADA	8	2.1
7	DOV	7	7	ADA	8	0.15295
8	DOV	8	10	ADA	11	9.650475
9	DOV	9	10	ADA	11	15.0581
10	DOV	10	10	ADA	11	27.576975
11	DOV	11	13	ADA	14	12.579275
12	DOV	12	13	ADA	14	15.50655
13	DOV	13	13	ADA	14	2.390275
14	DOV	14	16	ADA	17	0.1596
15	DOV	15	16	ADA	17	9.0828
16	DOV	16	16	ADA	17	10.4752
17	DOV	17	19	ADA	20	25.9548
18	DOV	18		ADA		13.1262

Figure 4-26: *Total Cargo* display showing total cargo shipped by one plan

The total number and weight of commodities to arrive at the selected airport are presented.

At the lowest level, the Channel Route Network problem is a cargo shipment problem, so it is essential to provide information about the cargo being shipped. Since the reports of *Total Cargo* and final cargo locations provide the same information in different formats, they are used similarly. Having all the cargo information in one place helps the user to determine the nature of the solution. For example, if a large amount of cargo is to be shipped to a particular base, then the user can expect to see many flights going to that particular base with a lot of cargo. The *Final Cargo Locations* display is especially helpful for this, because at the top it gives the total weight of the cargo that is destined for a particular airport. Similarly, seeing how much cargo is being shipped around the network as a whole is helpful. If a large amount of cargo is being shipped, then the user can expect to see many flights and missions with a lot of cargo.

The user can use these tools in conjunction with other tools. For example, imagine



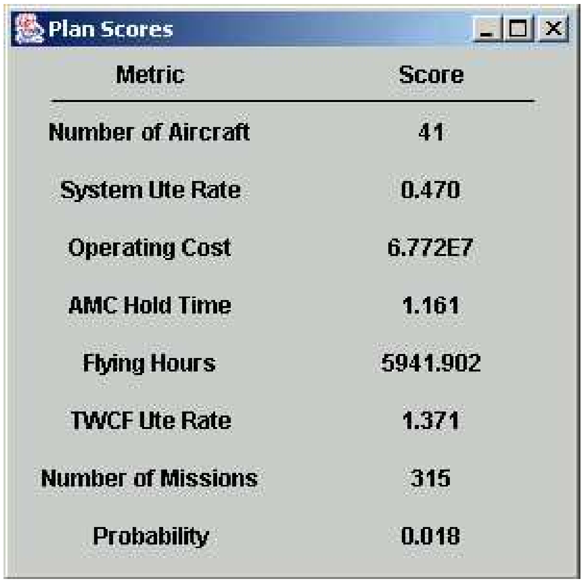
ADA

80 commodities, 490.97592 tons total

Commodity	Arrival Day	Origin	Ready Day	Departure Day	Weight
0	3	DOV	0	2	0.1
1	3	DOV	1	2	11.4
2	5	DOV	2	4	13.1
3	5	DOV	3	4	32.4
4	5	DOV	4	4	10.9
5	8	DOV	5	7	18.2
6	8	DOV	6	7	2.1
7	8	DOV	7	7	0.15295
8	11	DOV	8	10	9.650475
9	11	DOV	9	10	15.0581
10	11	DOV	10	10	27.576975
11	14	DOV	11	13	12.579275
12	14	DOV	12	13	15.50655
13	14	DOV	13	13	2.390275
14	17	DOV	14	16	0.1596

Figure 4-27: *Final Cargo Locations* display showing final planned cargo locations

that the user clicks on a flight leg in the *Individual Missions* display and sees a piece of cargo that is destined for DOV. This may lead him to wonder what other cargo is heading for DOV. He can find this information on either the *Total Cargo* or *Final Cargo Locations* display. The data in the *Total Cargo* and *Final Cargo Locations* reports can be sorted by columns, which allows the user to organize the information in many ways. He could sort the cargo information in order to quickly answer questions about what cargo is leaving a particular airport, what cargo is ready for shipment on a particular day, when cargo departs from its origin airport, when cargo arrives at its destination airport, and cargo weight using these displays. These displays are important because it allows the user to verify that all the cargo is ending up in the right place in a satisfactory amount of time.



Metric	Score
Number of Aircraft	41
System Ute Rate	0.470
Operating Cost	6.772E7
AMC Hold Time	1.161
Flying Hours	5941.902
TWCF Ute Rate	1.371
Number of Missions	315
Probability	0.018

Figure 4-28: *Plan Scores* report for a plan

4.4.7 Plan Scores

The metrics listed in Table 4.1 are the primary means of rating a plan. Clicking on the *Plan Scores* option from either the *Pop Up Menu* (Figure 4-12) or the *Options Selector* (Figure 4-14) displays a report that lists the scores of the plan for those metrics, as well as the probability of having the set of resources that was used to create the plan (Figure 4-28).

The information presented here is most useful when compared to same information for other plans. This capability is also given in the *Five-Dimensional Graph* (Section 4.3.1). However, listing the scores separately is also useful in case the user wishes to verify the plan's score for a metric. It is also useful in that it lists the exact numeric scores, whereas viewing the scores in the *Five-Dimensional Graph* is more qualitative and comparison-based. This display is the only place where all seven of the metric scores are visible at once. It allows the user to view information about the three metric scores that are not shown on the *Five-Dimensional Graph*. It also provides the exact values of the metrics plotted on the graph. Multiple plan score reports can be viewed simultaneously, allowing the user to compare multiple plans at once. This comparison is more quantitative than that provided by the *Five-Dimensional*

Graph. In addition, one display of *Plan Scores* can be used alone to view the exact metric scores for a plan.

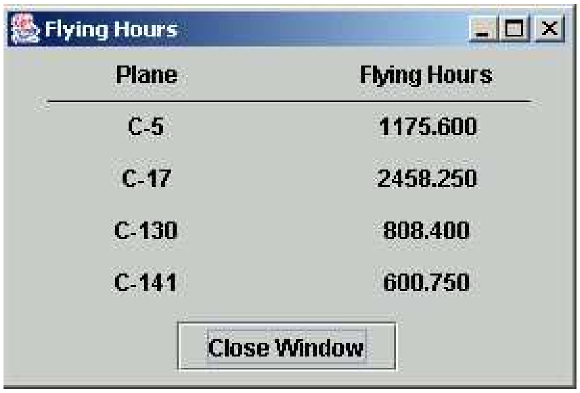
4.4.8 Flying Hours

The Flying Hours Program is an important component of the training program of the military. Channel route missions are a major source of flying experience for pilots. Therefore, it is important to show just how many flying hours will be gained as a result of executing the plan. Clicking on the *Flying Hours* option from either the *Pop Up Menu* (Figure 4-12) or the *Options Selector* (Figure 4-14) creates a window with information about the Flying Hour Requirements (Figure 4-29). For each plane type, the number of actual Flying Hours in the current plan is listed.

The information contained in this display is more specific than the Flying Hours metric in that it deals with individual aircraft types instead of all aircraft as a single set. It can be used in conjunction with the listing of metric scores to answer questions about the number of Flying Hours in the plan. For example, imagine that the user knows that they are behind the requirement for C17 flying hours and that they need to catch up this month. The listing of metric scores shows how many Flying Hours are in the plan as a whole, but it does not show how many hours are provided for each plane type. This information is found on the *Flying Hours* display. Here the user can see exactly how many hours are present in the plan for each plane type. This information will allow the user to determine whether this plan uses enough flying hours for C17 planes to catch up to the cumulative flying hours requirement.

4.4.9 Help

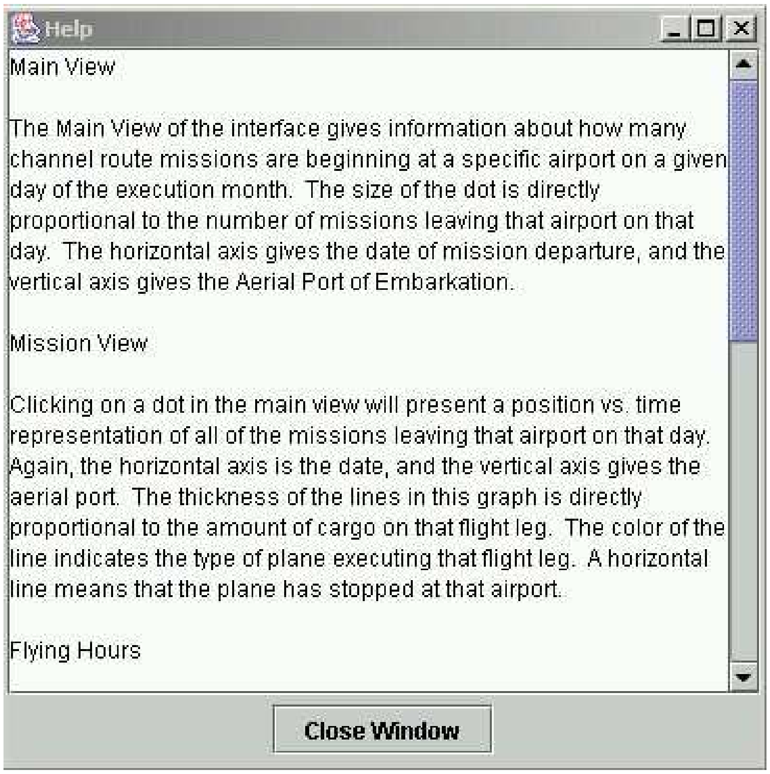
Accessing the *Help* option from either the *Pop Up Menu* (Figure 4-12) or the *Options Selector* (Figure 4-14) creates a window with a paragraph about each of the information windows (Figure 4-30). It is simply a window with text and a close button at the bottom. The *Help* display reminds the user what tools are available and how to use them. It can be accessed at any time and is an alternative to referring to a



Plane	Flying Hours
C-5	1175.600
C-17	2458.250
C-130	808.400
C-141	600.750

Figure 4-29: *Flying Hours* report showing flying hours provided by the plan

printed source, such as this thesis, for information about the tools in ChRIS.



Main View

The Main View of the interface gives information about how many channel route missions are beginning at a specific airport on a given day of the execution month. The size of the dot is directly proportional to the number of missions leaving that airport on that day. The horizontal axis gives the date of mission departure, and the vertical axis gives the Aerial Port of Embarkation.

Mission View

Clicking on a dot in the main view will present a position vs. time representation of all of the missions leaving that airport on that day. Again, the horizontal axis is the date, and the vertical axis gives the aerial port. The thickness of the lines in this graph is directly proportional to the amount of cargo on that flight leg. The color of the line indicates the type of plane executing that flight leg. A horizontal line means that the plane has stopped at that airport.

Flying Hours

Figure 4-30: *Help* tool

Chapter 5

Example of System Use

This chapter provides an example of a user interacting with ChRIS. The user in this example is a human planner at AMC trying to select the best plan from the set of plans created by the Channel Route Network Planning System (CRNPS) . Figure 5-1 is a flowchart of the total interaction once the system is started. The interaction starts at the top and traverses the chart like a depth first search with backtracking until the plan is accepted. To move past a terminal that is not the accepted plan, one should backtrack to the last branching point and follow the next option.

The planner determines a set of probable input values for the optimization. He runs the optimization for all of the different possible combinations of input values, producing a plan for each one. For a typical set of input values, in its current implementation, running the optimization repeatedly takes approximately 1 day and creates approximately 1000 plans. Once the optimization has stopped running, he can begin to view the output. The task of the user is to look at the output and choose a plan for execution.

When he starts the system, he is shown the *Metric Selection Tool* (Figure 5-2). At first he chooses to plot Aircraft Used along the x axis, Operating Cost along the y axis, Missions Flown using dot color, and TWCF Ute Rate using dot size (Figure 5-3). Upon viewing this graph, he sees that the points are spread out evenly along the x axis. This makes him think that Aircraft Used is not informative and not important enough to him to be plotted and that TWCF Ute Rate is too important to be plotted

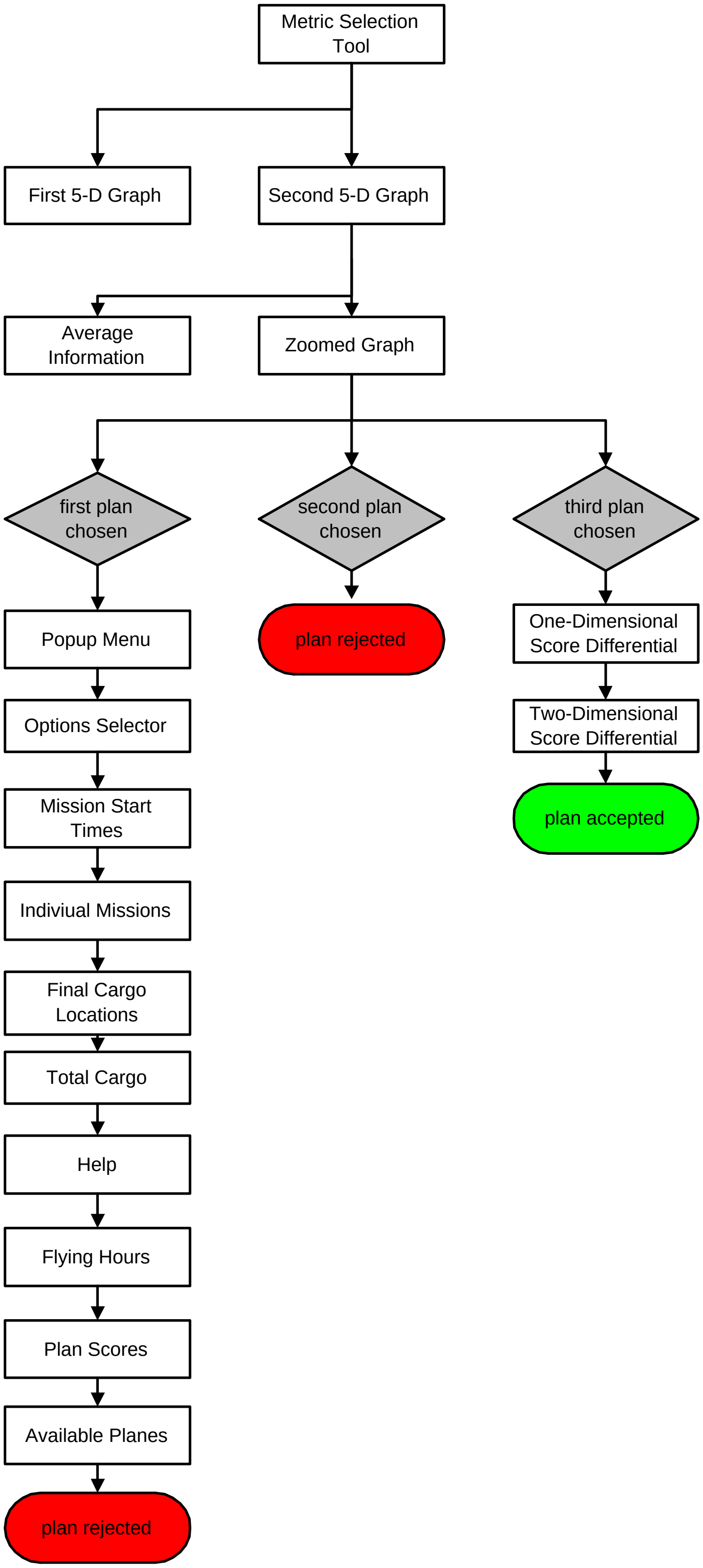


Figure 5-1: Flowchart of example interaction

X Axis
Aircraft Used

Y Axis
Operating Cost

Dot Color
Missions Flown

Dot Size
TWCF Ute Rate

Figure 5-2: Initial *Metric Selection Tool* viewed by user

using dot size.

Going back to the *Metric Selection Tool* , he chooses to plot Operating Cost along the x axis, TWCF Ute Rate along the y axis, AMC Hold Time using dot color, and Missions Flown using dot size (Figure 5-4). Based upon the objectives of the chosen metrics, to find a plan that has a low Operating Cost, high TWCF Utilization Rate, low AMC Hold Time, and low number of Missions Flown, he would want to look for a small green dot in the upper left corner of this graph.

There are two clusters in the graph: big green and blue-green dots in the lower right corner of the graph, and small red and purple dots in the upper left of the graph. After selecting each of the clusters by drawing a rectangle around it (Figure 5-5), he uses the averages button on the *Legend* for the *Five-Dimensional Graph* to find out its overall character (Figure 5-6). He notices that the numbers of planes available for each cluster are very similar. However, the plans in the lower right corner fly many more missions, and have lower TWCF Utilization Rates and AMC Hold Times. They also cost significantly more, have more flying hours, and use more aircraft. Having seen this information, he decides TWCF Utilization Rate, Operating Cost, and Missions Flown are more important than AMC Hold Time, so he looks for a small dot in the upper left of the graph, paying no attention to color.

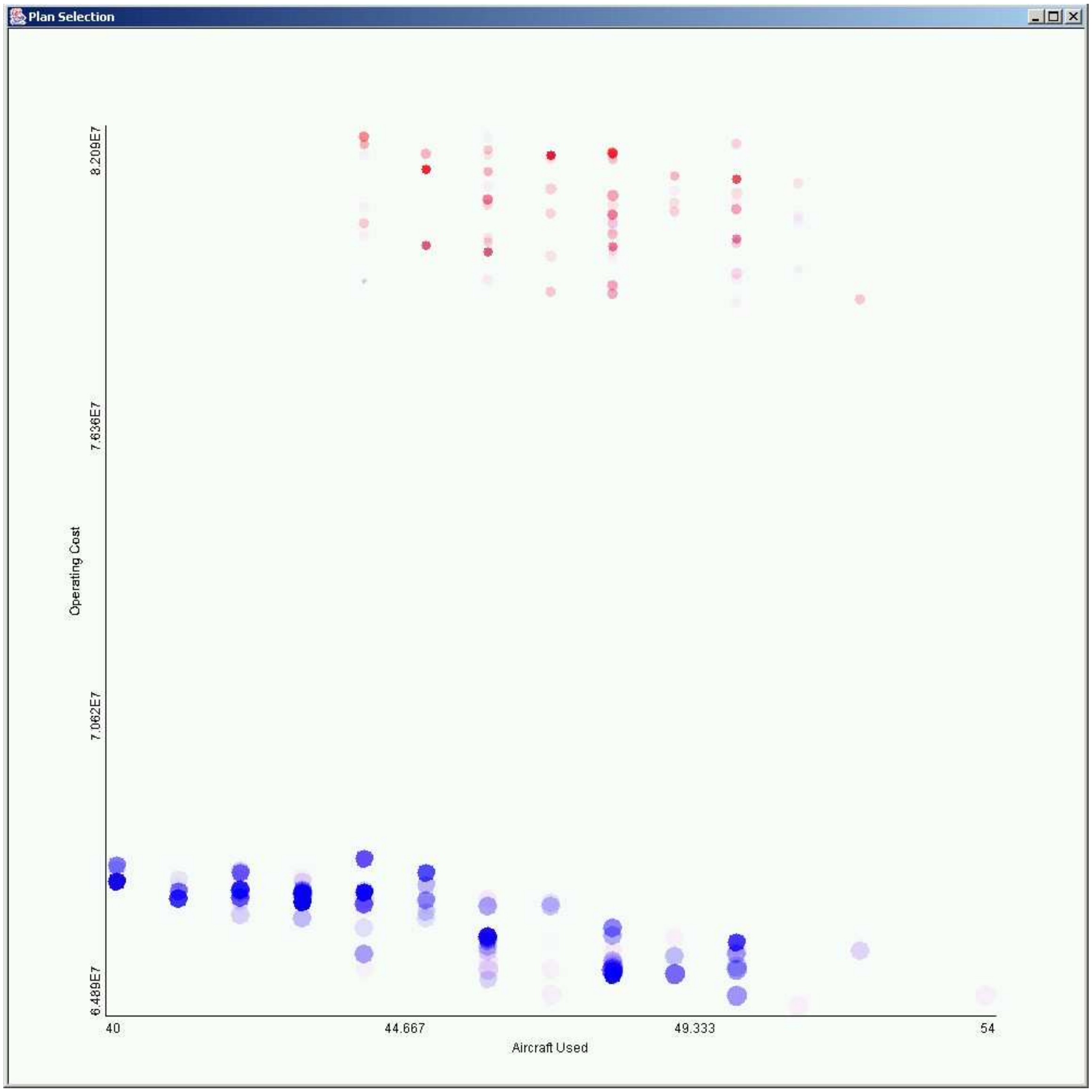


Figure 5-3: Initial *Five-Dimensional Graph* viewed by user. The axes are Aircraft Used (x), Operating Cost(y), Missions Flown (color), and TWCF Ute Rate (size).

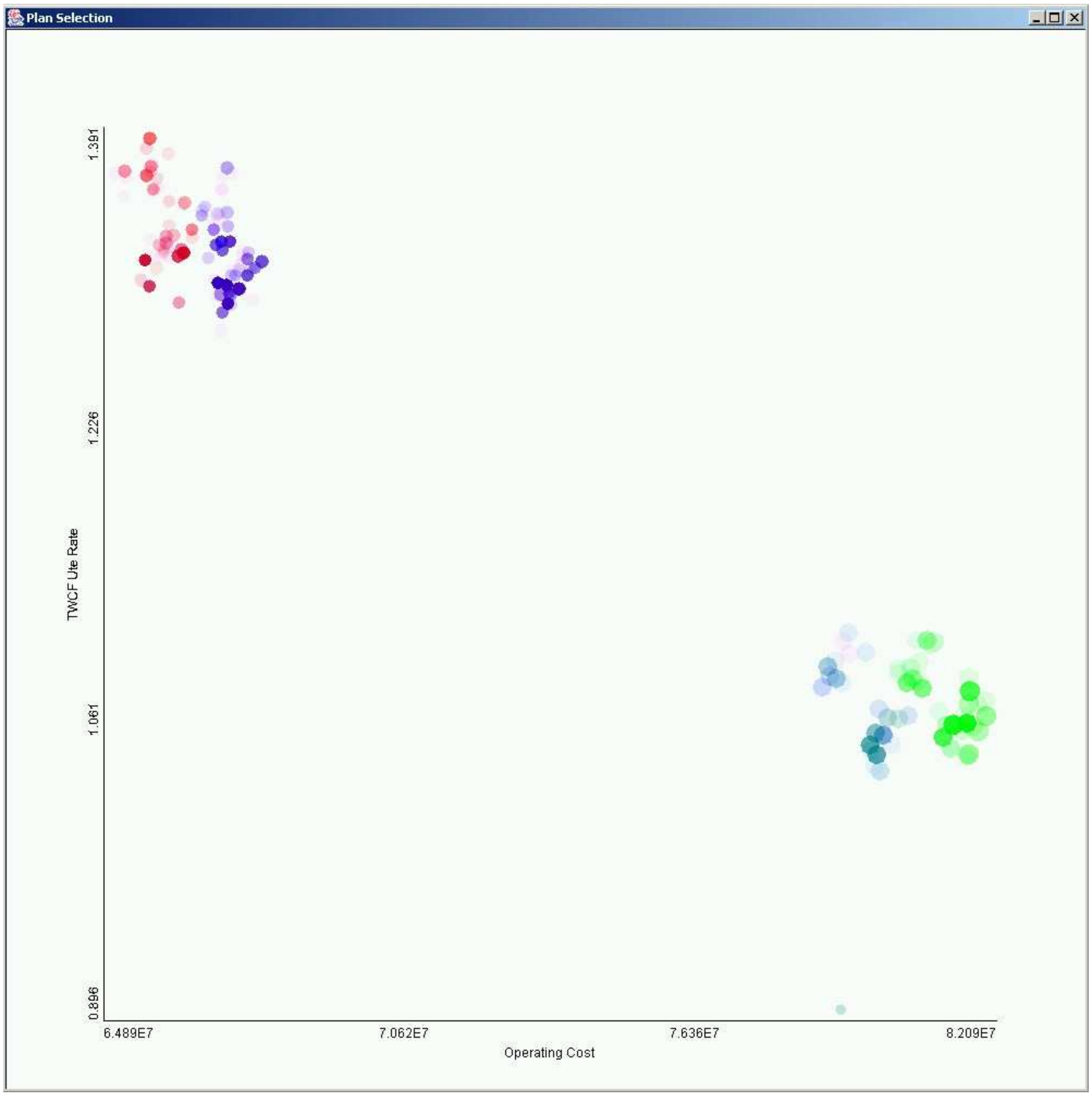


Figure 5-4: Second *Five-Dimensional Graph* viewed by user. The axes are Operating Cost(x), TWCF Ute Rate (y), AMC Hold Time (color), and Missions Flown (size).

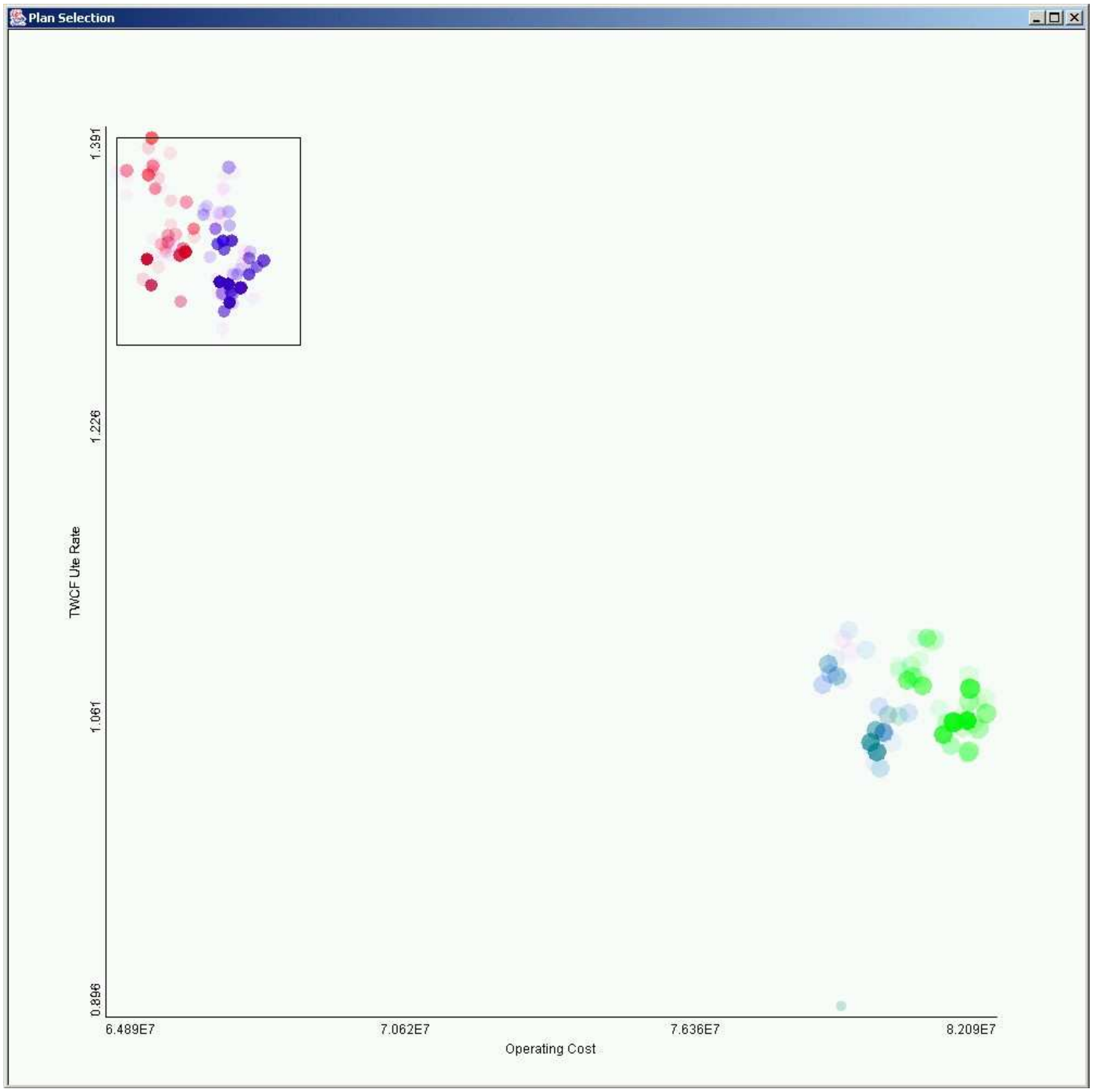
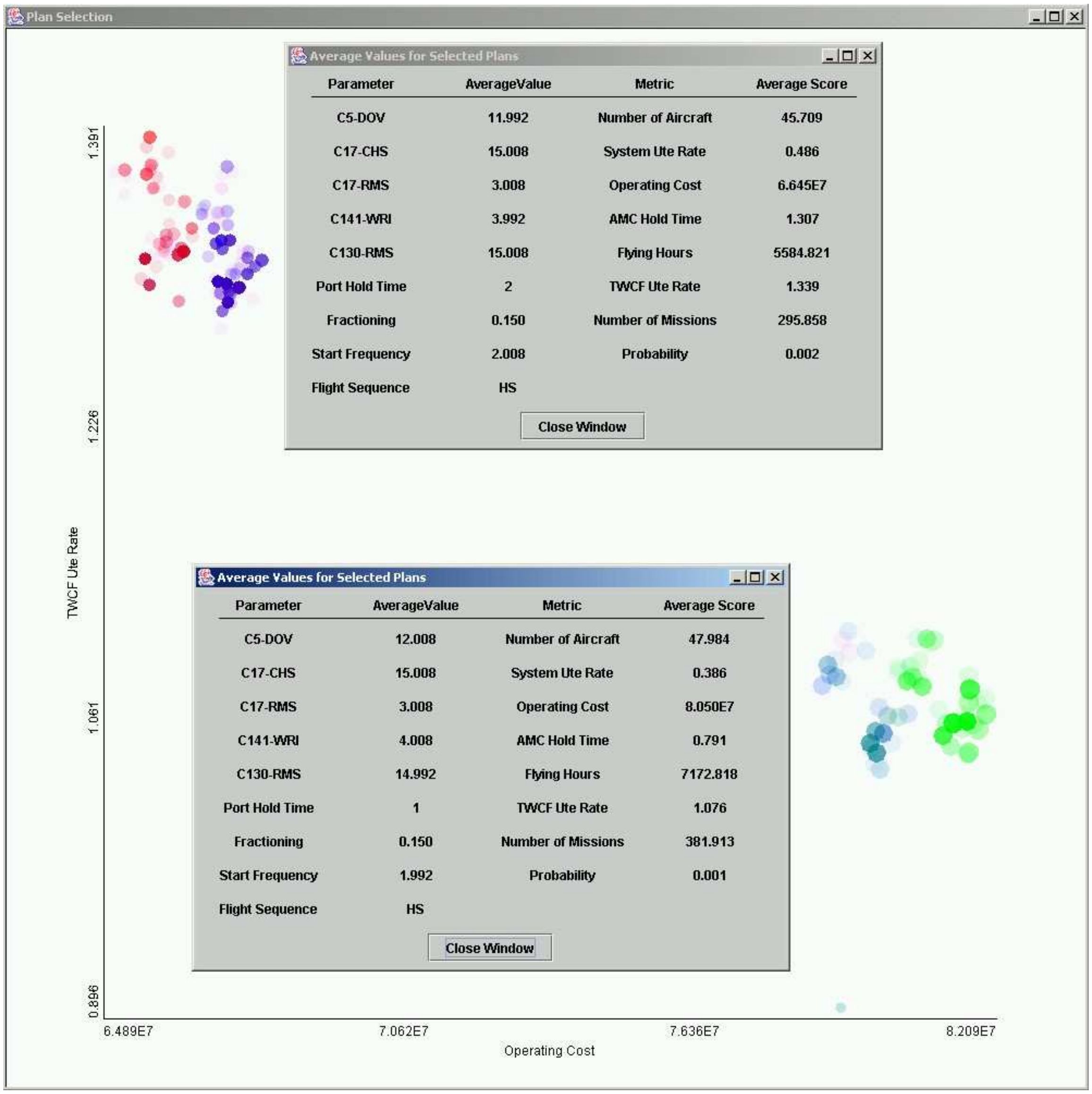


Figure 5-5: Selection of upper left cluster



Parameter	AverageValue	Metric	Average Score
C5-DOV	11.992	Number of Aircraft	45.709
C17-CHS	15.008	System Ute Rate	0.486
C17-RMS	3.008	Operating Cost	6.645E7
C141-WRI	3.992	AMC Hold Time	1.307
C130-RMS	15.008	Flying Hours	5584.821
Port Hold Time	2	TWCF Ute Rate	1.339
Fractioning	0.150	Number of Missions	295.858
Start Frequency	2.008	Probability	0.002
Flight Sequence	HS		

Parameter	AverageValue	Metric	Average Score
C5-DOV	12.008	Number of Aircraft	47.984
C17-CHS	15.008	System Ute Rate	0.386
C17-RMS	3.008	Operating Cost	8.050E7
C141-WRI	4.008	AMC Hold Time	0.791
C130-RMS	14.992	Flying Hours	7172.818
Port Hold Time	1	TWCF Ute Rate	1.076
Fractioning	0.150	Number of Missions	381.913
Start Frequency	1.992	Probability	0.001
Flight Sequence	HS		

Figure 5-6: Average information for two clusters

In order to get a better view of the cluster and to remove from view those plans in which he is no longer interested, he selects this region of the graph and zooms in on it. At first, he zooms in too far and accidentally makes some of the plans on the border of the cluster (Figure 5-7) invisible. Wanting to try again and include those currently invisible plans, he clicks the Reset button on the *Legend* to reset the graph. This time he does not zoom in enough, leaving too much space around the cluster (Figure 5-8), and he zooms in one more time to get the desired region (Figure 5-9).

He chooses the plan that is closest to the upper left corner (Figure 5-10). He uses the *Legend* to estimate that the AMC Hold Time is 1.39, the number of Missions Flown is 300, and the probability for the plan is .003.

Next he right-clicks on the point, making the *Pop Up Menu* for that plan visible. First, he views the identifying information, just to get an idea of what plan he is looking at (Figure 5-11). Then he decides to view the detailed displays for that plan, so he clicks on the *Options Selector* option in the Options submenu of the *Pop Up Menu* (Figure 5-12). This brings up the *Options Selector* (Figure 5-13), which he uses to access the other pieces of information for that plan.

First, he looks at the information about the mission start times and the actual missions. He does this by bringing up the display of *Mission Start Times* (Figure 5-14) with the appropriate button on the *Options Selector*. It strikes him that RMS has at least three missions leaving on most days, while the other departure airports are not as heavily used. He also notices that CHS has a repetitive structure, with 5 or 7 missions leaving every three days. Clicking on several of these dots, he finds that the graphs of *Individual Missions* leaving those airports on those days (i.e. Figure 5-15) are very similar in appearance, which means that the actual missions are very similar in structure.

He then looks more closely at the display of missions leaving CHS on day 2 (Figure 5-15). The green color of the missions indicates that all of these missions are flown by C17 planes based at CHS, according to the legend. The first thing he does is to click along the flight legs of the mission that visits AVB in order to determine what and how cargo is being shipped. After clicking on each flight leg in the mission,

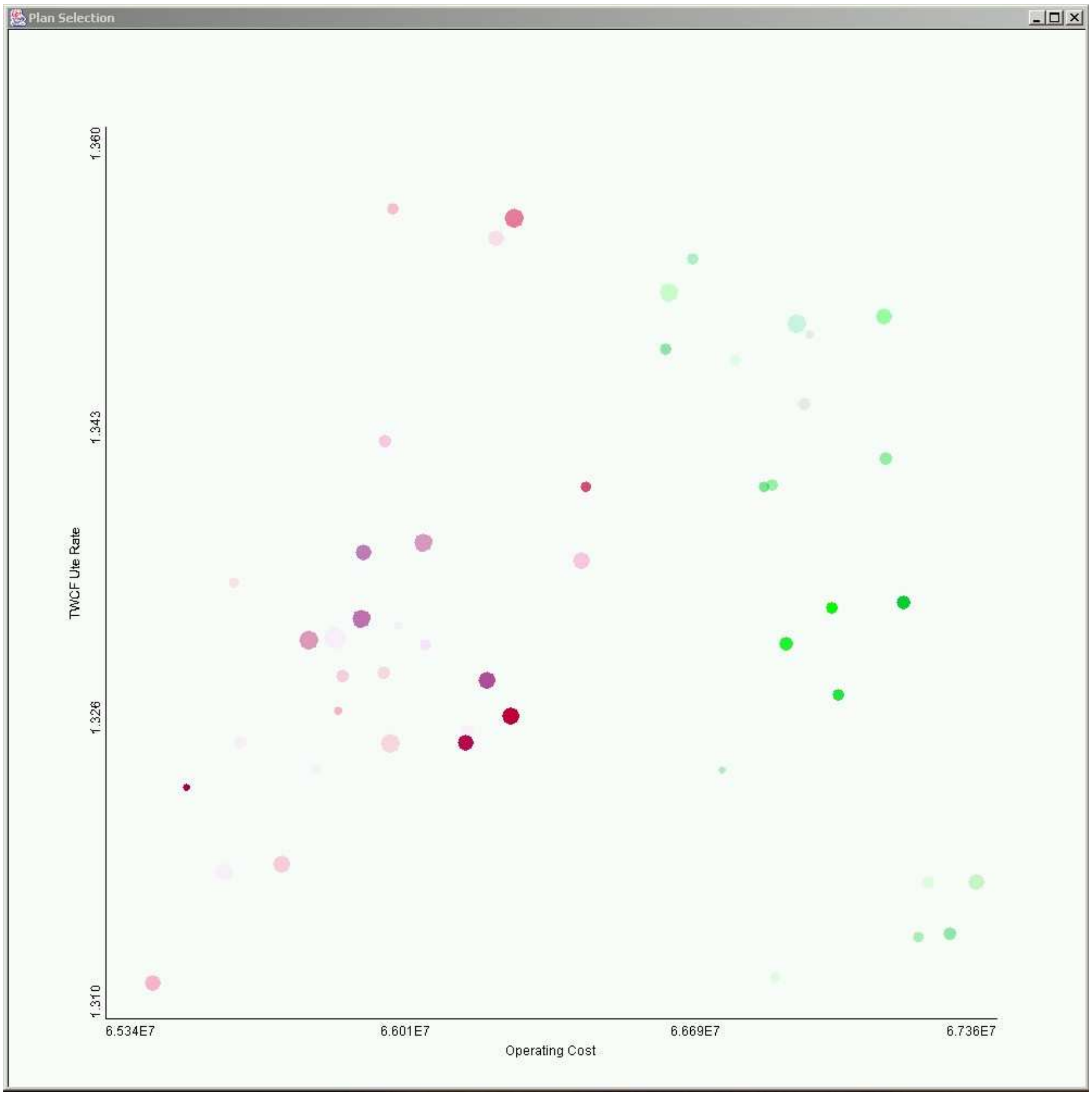


Figure 5-7: View of top left corner of Figure 5-4 when zoomed in too far

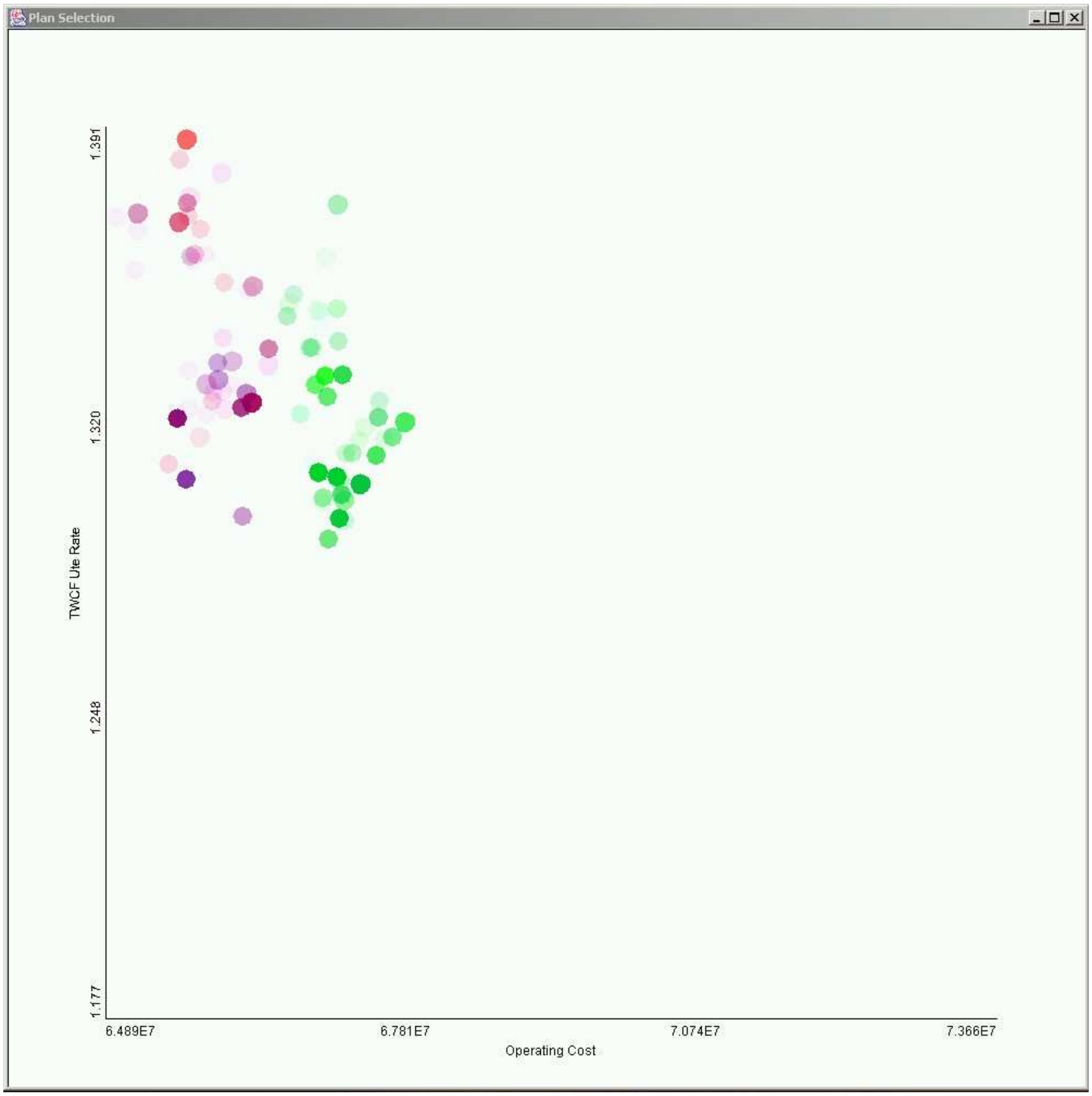


Figure 5-8: View of top left corner of Figure 5-4 when not zoomed in enough

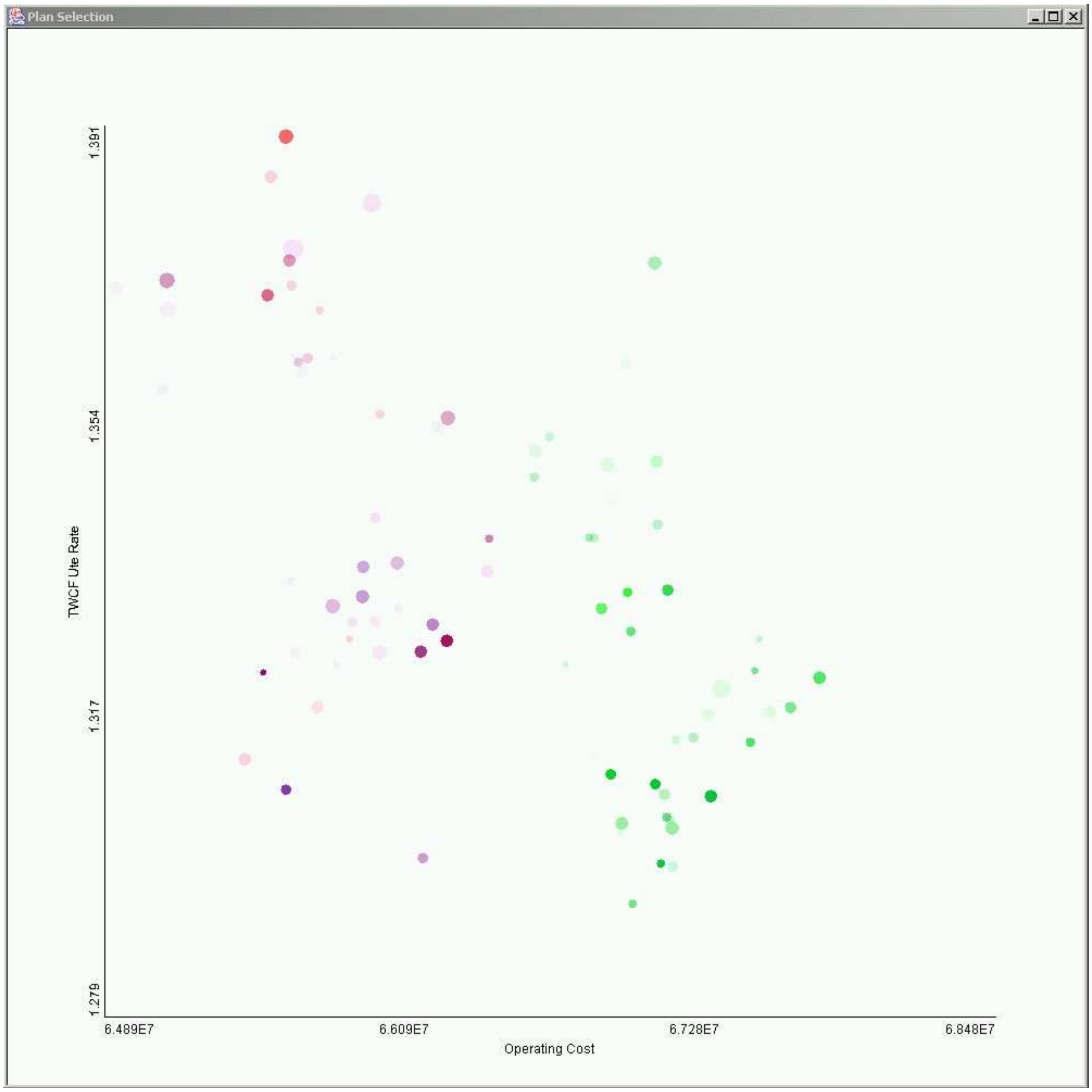


Figure 5-9: Correctly zoomed in view of top left corner of Figure 5-4

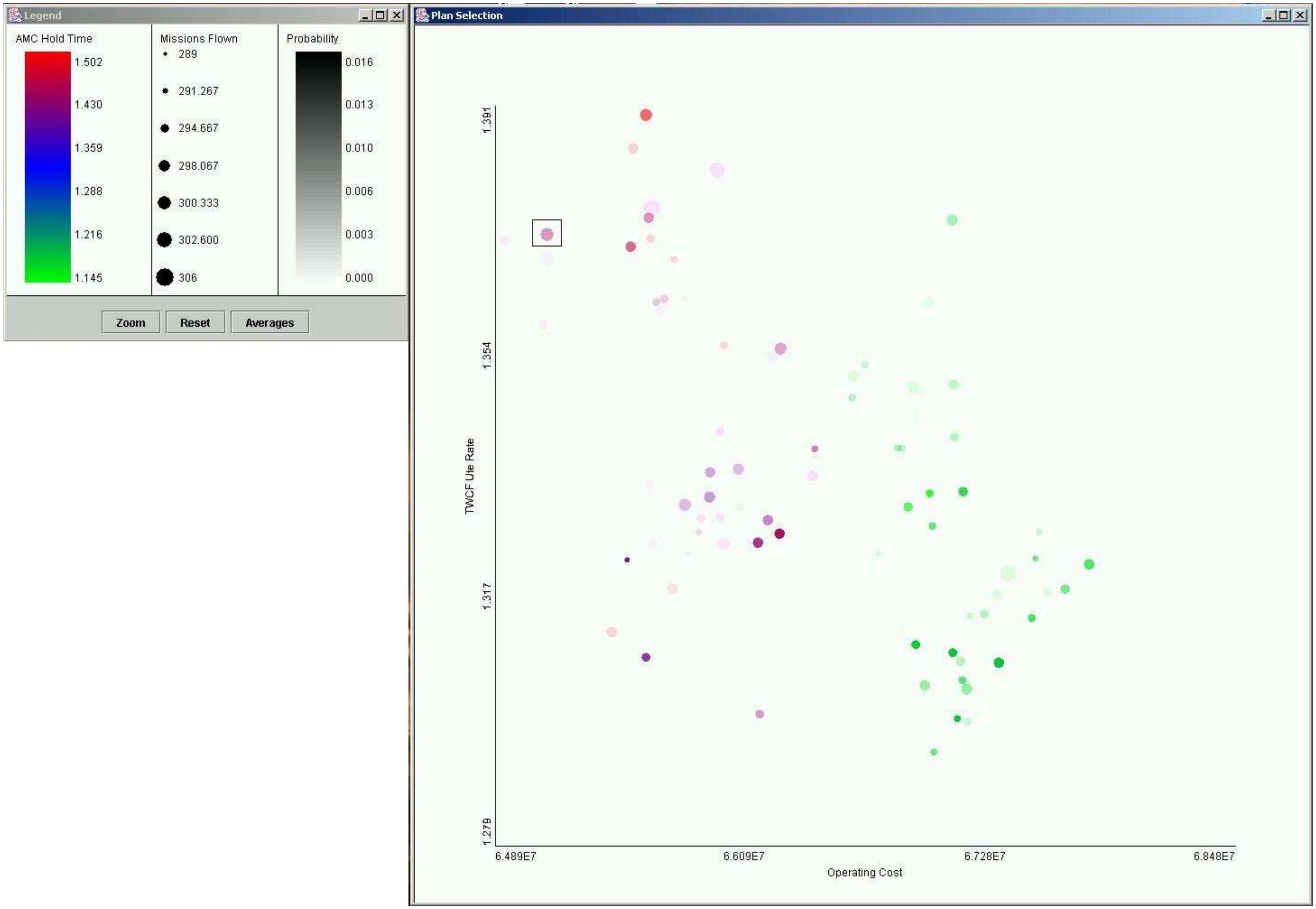


Figure 5-10: First plan explored by user. The first plan explored by the user is represented by the point inside the box.

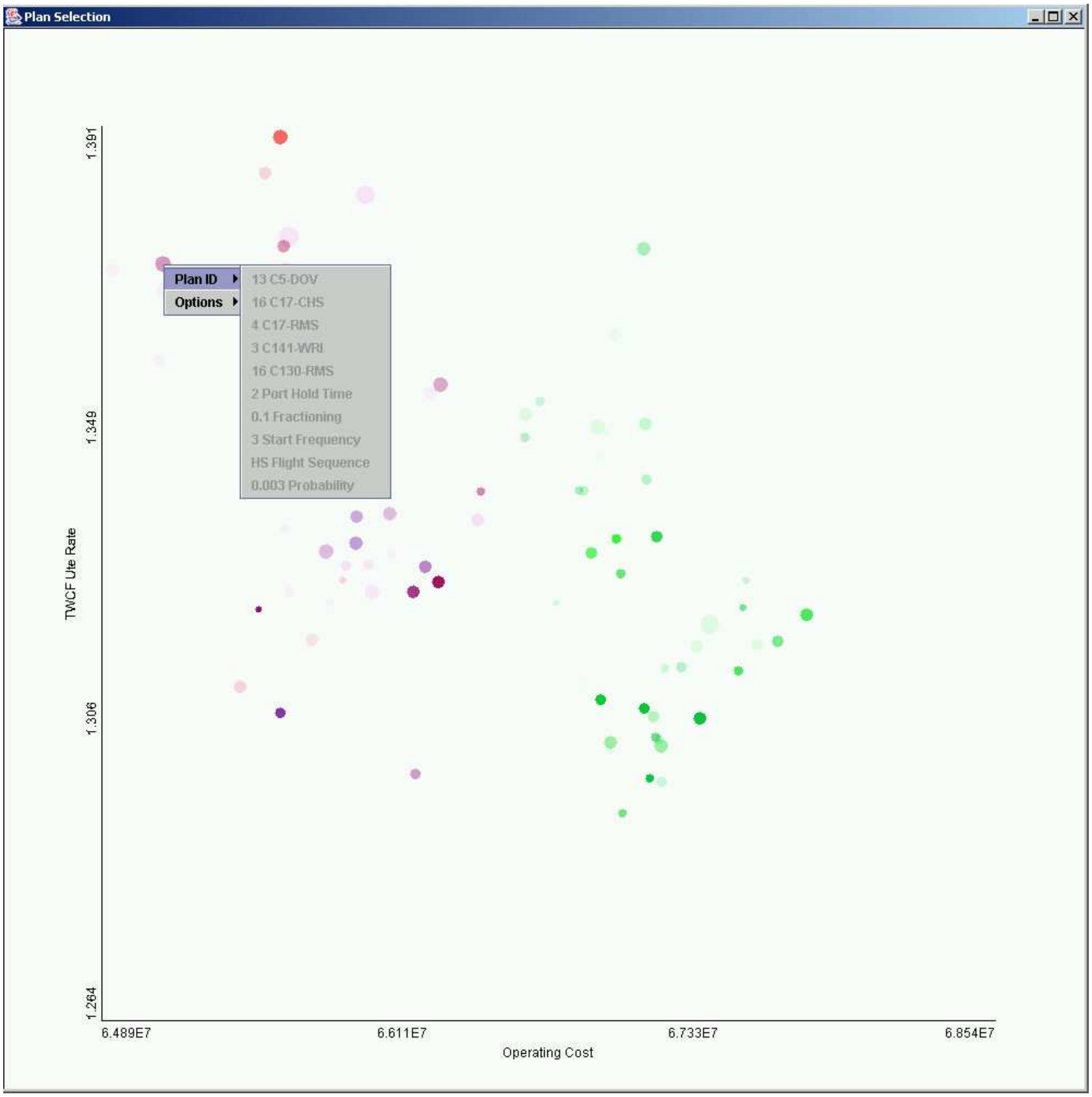


Figure 5-11: Identifying information *Pop Up Menu* for first plan explored by user

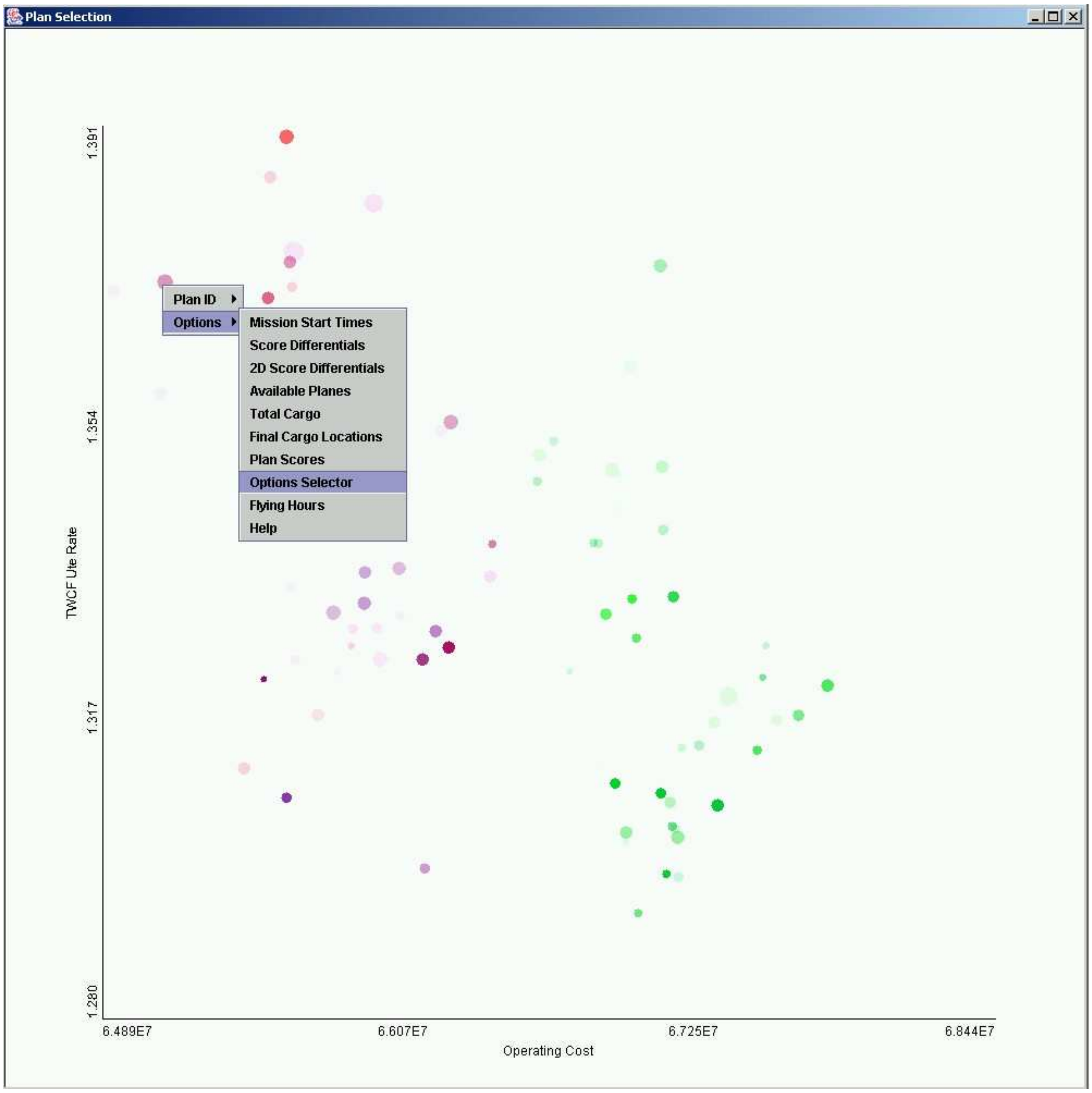
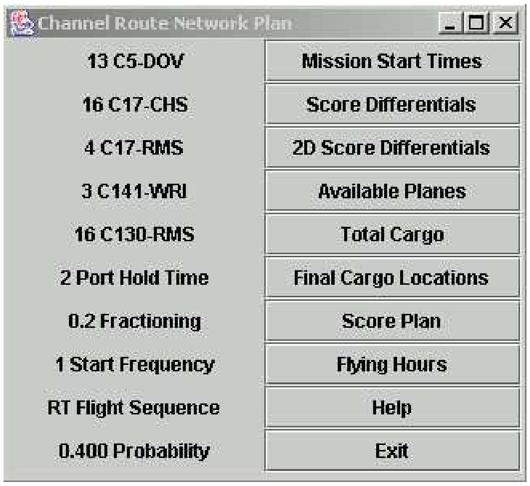


Figure 5-12: Selection of *Options Selector* from options *Pop Up Menu* for first plan explored by user



13 C5-DOV
16 C17-CHS
4 C17-RMS
3 C141-WRI
16 C130-RMS
2 Port Hold Time
0.2 Fractioning
1 Start Frequency
RT Flight Sequence
0.400 Probability

Figure 5-13: *Options Selector* for first plan explored by user

he views the list of cargo on that flight leg in the legend. For example, the list cargo in Figure 5-15 appeared after the user clicked on the flight leg between RMS and AVB. From the listing of cargo for that flight leg in the legend, he can tell that some of the cargo originally came from DOV, and the rest came from RMS. The changes in thickness of the lines indicate that cargo is dropped off at AVB, and this is verified by the fact that all of the cargo listed in the graph legend is destined for AVB. He notes that this display only provides information on missions originating at CHS on day 2, and he wonders what other cargo ends up at AVB.

At this point, he goes back to the *Options Selector* (Figure 5-13) and uses the appropriate button to access the *Final Cargo Locations* report (Figure 5-16). After selecting AVB from the drop down select box at the top, he sees that 48 pieces of cargo end up there, and about half come from DOV, with the other half coming from RMS. He notices that some of the cargo takes 3 days from when it is ready to reach its destination. He thinks this is a problem, but not a big enough one to make him want to switch plans. He then uses the Options Selector to access the *Total Cargo* display (Figure 5-17), wanting information about all the cargo that leaves DOV. He clicks on the header of the origin column to sort the cargo by its origin airport, and he sees that approximately one third of all cargo leaves from DOV.

Next, the user returns to the display of missions leaving CHS on day 2 (Figure 5-

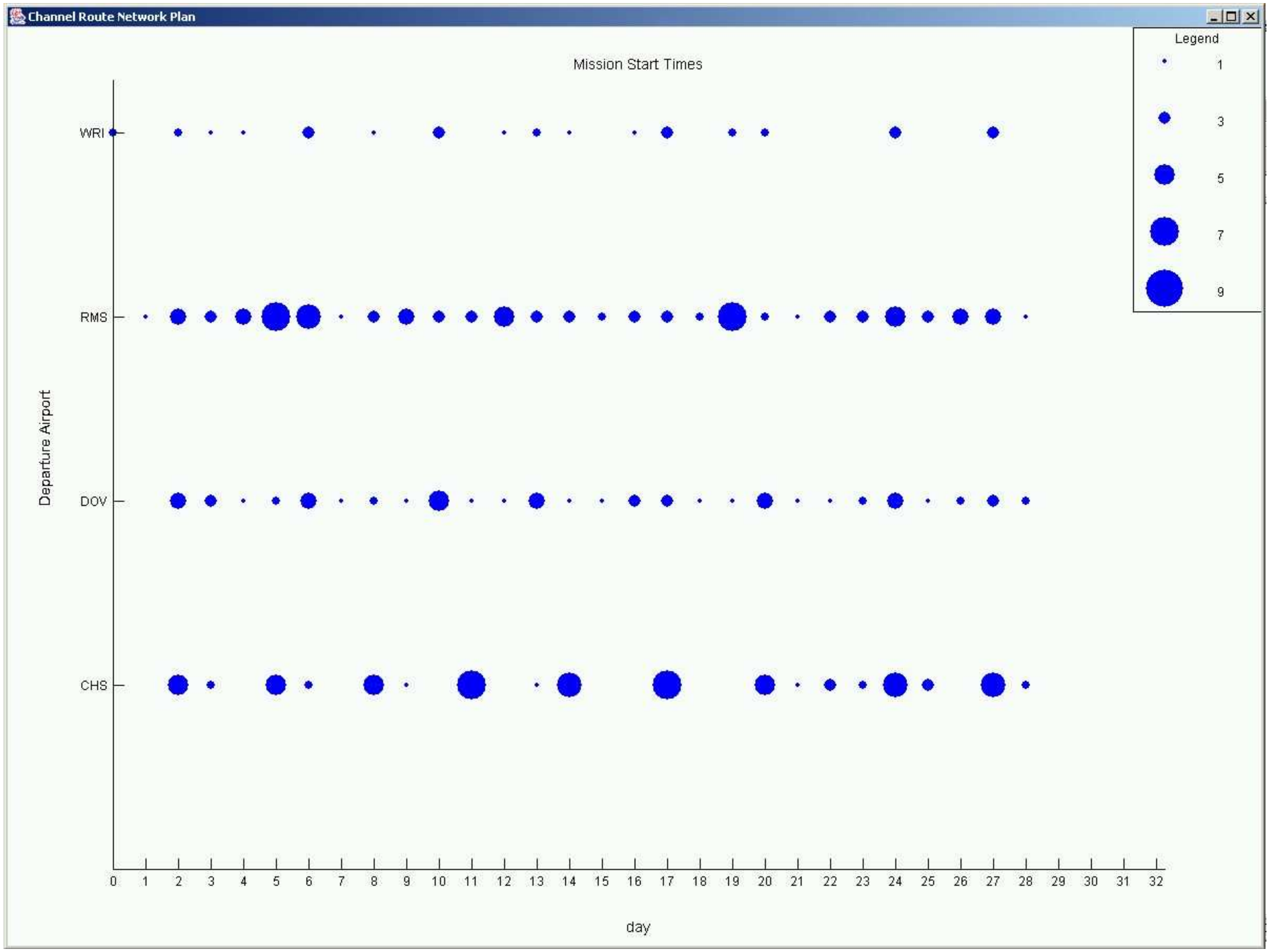
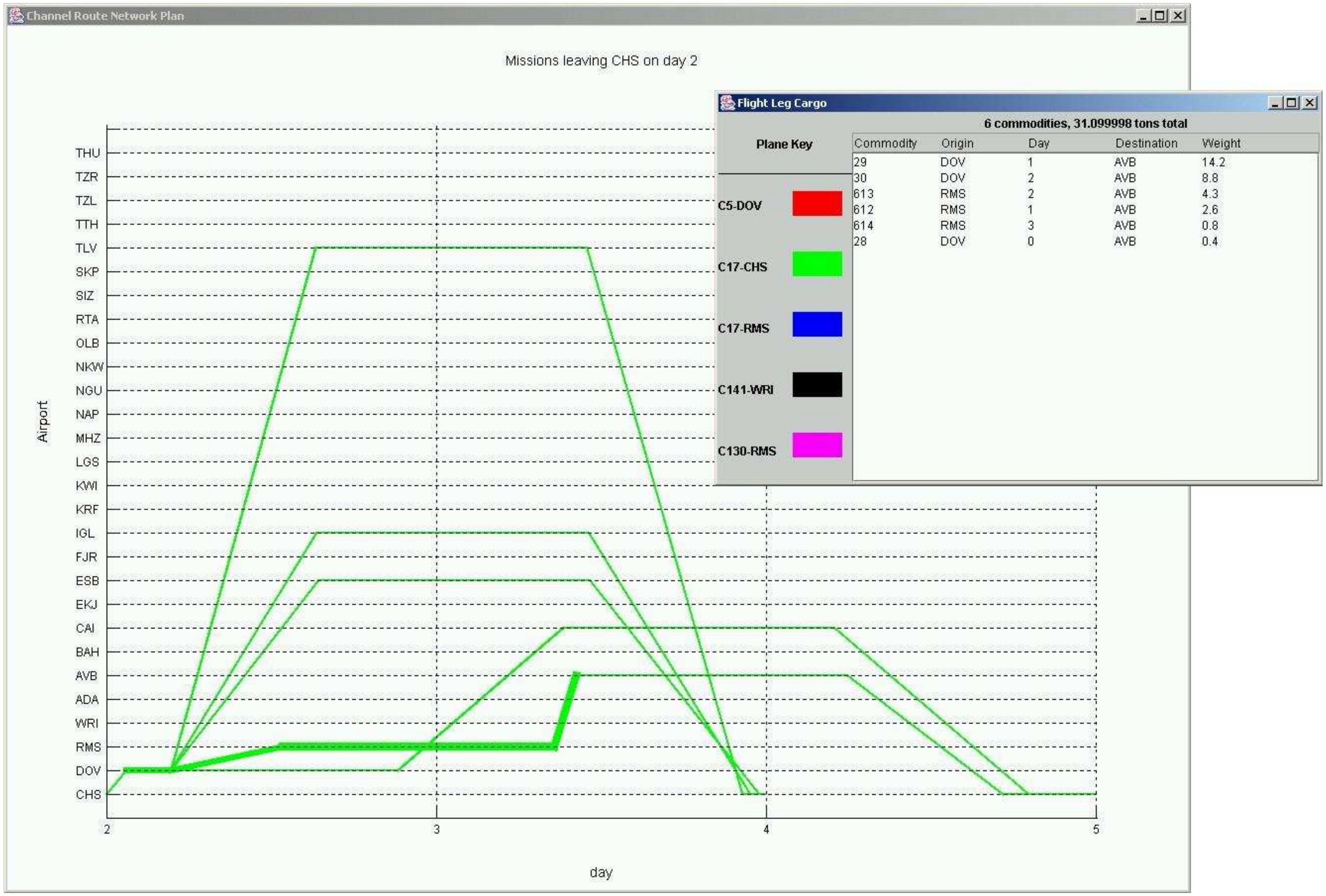


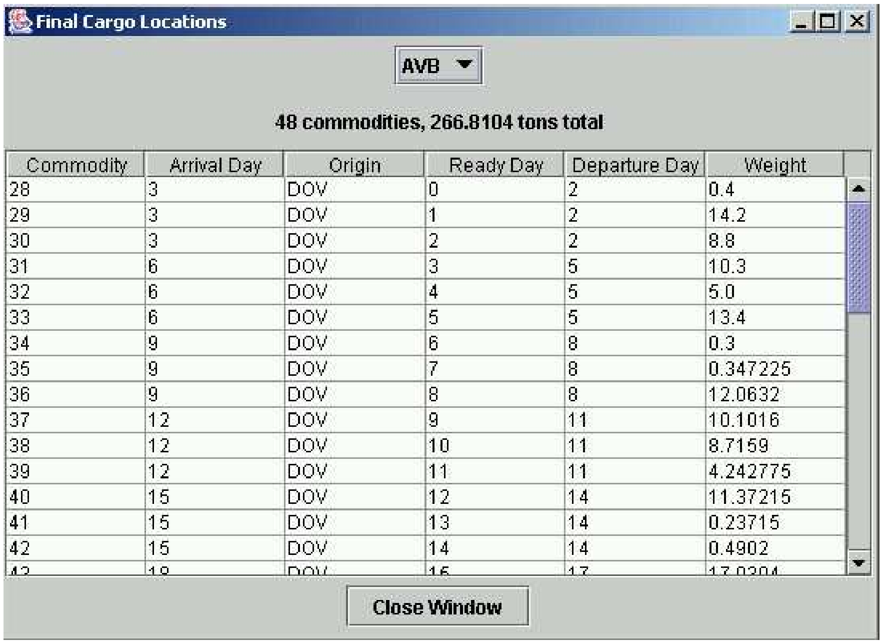
Figure 5-14: *Mission Start Times* for first plan explored by user. Each dot is directly proportional to the number of missions leaving that airport on that day.



6 commodities, 31.099998 tons total

Commodity	Origin	Day	Destination	Weight
29	DOV	1	AVB	14.2
30	DOV	2	AVB	8.8
613	RMS	2	AVB	4.3
612	RMS	1	AVB	2.6
614	RMS	3	AVB	0.8
28	DOV	0	AVB	0.4

Figure 5-15: Display of *Individual Missions* leaving CHS on day 2 for first plan explored by user. Each line represents a flight leg in a mission. The thickness of the line represents the amount of cargo on that flight leg. The legend lists the cargo carried on the flight leg between RMS and AVB.



48 commodities, 266.8104 tons total

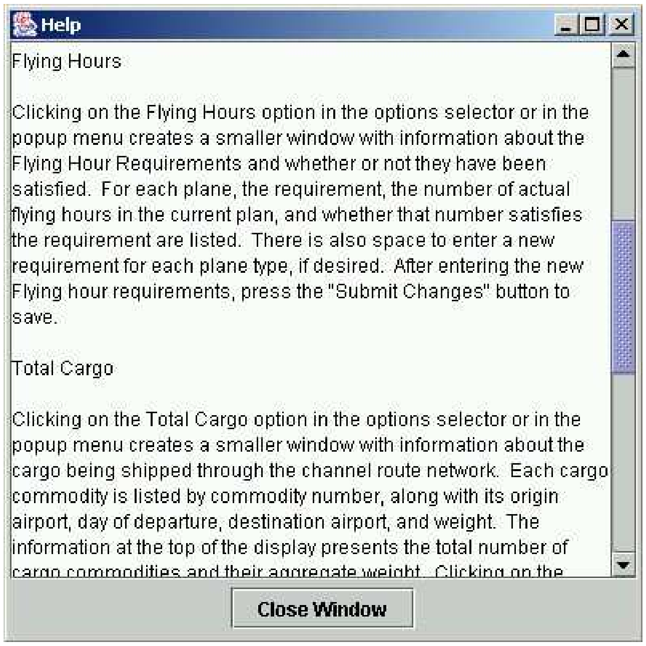
Commodity	Arrival Day	Origin	Ready Day	Departure Day	Weight
28	3	DOV	0	2	0.4
29	3	DOV	1	2	14.2
30	3	DOV	2	2	8.8
31	6	DOV	3	5	10.3
32	6	DOV	4	5	5.0
33	6	DOV	5	5	13.4
34	9	DOV	6	8	0.3
35	9	DOV	7	8	0.347225
36	9	DOV	8	8	12.0632
37	12	DOV	9	11	10.1016
38	12	DOV	10	11	8.7159
39	12	DOV	11	11	4.242775
40	15	DOV	12	14	11.37215
41	15	DOV	13	14	0.23715
42	15	DOV	14	14	0.4902

Figure 5-16: *Final Cargo Locations* report for first plan explored by user

1016 commodities, 13493.891 tons total

Commodity	Origin	Ready Day	Departure Day	Destination	Arrival Day	Weight
0	DOV	0	2	ADA	3	0.1
1	DOV	1	2	ADA	3	11.4
2	DOV	2	4	ADA	5	13.1
3	DOV	3	4	ADA	5	32.4
4	DOV	4	4	ADA	5	10.9
5	DOV	5	7	ADA	8	18.2
6	DOV	6	7	ADA	8	2.1
7	DOV	7	7	ADA	8	0.15295
8	DOV	8	10	ADA	11	9.650475
9	DOV	9	10	ADA	11	15.0581
10	DOV	10	10	ADA	11	27.576975
11	DOV	11	13	ADA	14	12.579275
12	DOV	12	13	ADA	14	15.50655
13	DOV	13	13	ADA	14	2.390275
14	DOV	14	16	ADA	17	0.1596
15	DOV	15	16	ADA	17	9.0828
16	DOV	16	16	ADA	17	10.4752
17	DOV	17	19	ADA	20	25.9548
18	DOV	18	19	ADA	20	13.1262

Figure 5-17: *Total Cargo* display for first plan explored by user



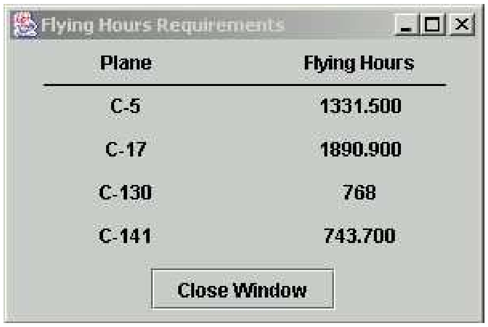
Flying Hours

Clicking on the Flying Hours option in the options selector or in the popup menu creates a smaller window with information about the Flying Hour Requirements and whether or not they have been satisfied. For each plane, the requirement, the number of actual flying hours in the current plan, and whether that number satisfies the requirement are listed. There is also space to enter a new requirement for each plane type, if desired. After entering the new Flying hour requirements, press the "Submit Changes" button to save.

Total Cargo

Clicking on the Total Cargo option in the options selector or in the popup menu creates a smaller window with information about the cargo being shipped through the channel route network. Each cargo commodity is listed by commodity number, along with its origin airport, day of departure, destination airport, and weight. The information at the top of the display presents the total number of cargo commodities and their aggregate weight. Clicking on the

Figure 5-18: *Help* tool viewed by user

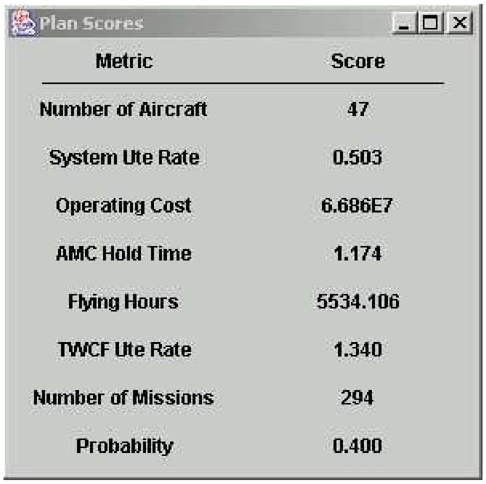


Plane	Flying Hours
C-5	1331.500
C-17	1890.900
C-130	768
C-141	743.700

Figure 5-19: *Flying Hours* display for first plan explored by user

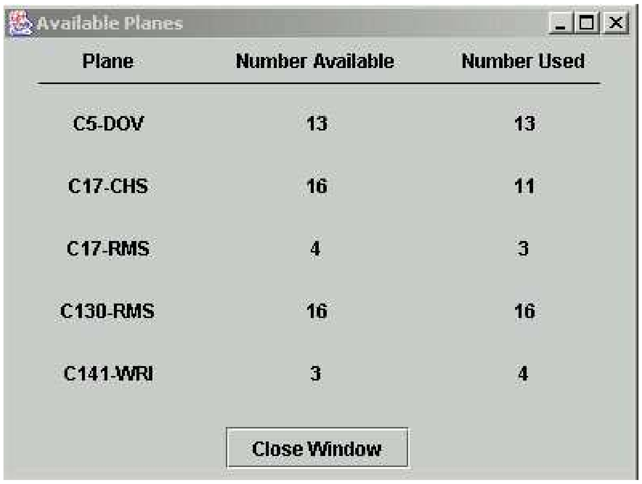
15). He notices that some of the missions take almost three days to complete. This makes him think that the majority of the flying hours gained in this plan are for C17 planes. However, he has forgotten whether information on the number of Flying Hours for a specific plane type is accessible from the system, so he accesses the *Help* tool (Figure 5-18) from the same Options Selector. This tells him how to get to the *Flying Hours* display. He does this, and realizes that C17 planes do have the largest number of flying hours (Figure 5-19).

The user then returns to the *Options Selector* to display the *Plan Scores* tool (Figure 5-20). He notices that 47 planes are being used, and this seems too high to



Metric	Score
Number of Aircraft	47
System Ute Rate	0.503
Operating Cost	6.686E7
AMC Hold Time	1.174
Flying Hours	5534.106
TWCF Ute Rate	1.340
Number of Missions	294
Probability	0.400

Figure 5-20: *Plan Scores* display for first plan explored by user



Plane	Number Available	Number Used
C5-DOV	13	13
C17-CHS	16	11
C17-RMS	4	3
C130-RMS	16	16
C141-WRI	3	4

Figure 5-21: *Available Planes* display for first plan explored by user

him. He checks the *Available Planes* display (Figure 5-21) and realizes that he wants a plan that uses fewer C5-DOV planes, because they are very expensive to fly.

He goes back to the first satisfactory *Five-Dimensional Graph* (Figure 5-4) to choose another plan. First, he uses the averages button to verify that there are plans with fewer planes. He knows that there are because the average number of planes used in the upper left cluster is less than 47. Next, he examines the plan that is the next closest to the upper left corner (Figure 5-22), but decides not to consider it any further, because its probability is very low. The chances of the resources needed to execute this plan being available are too low to warrant further study.

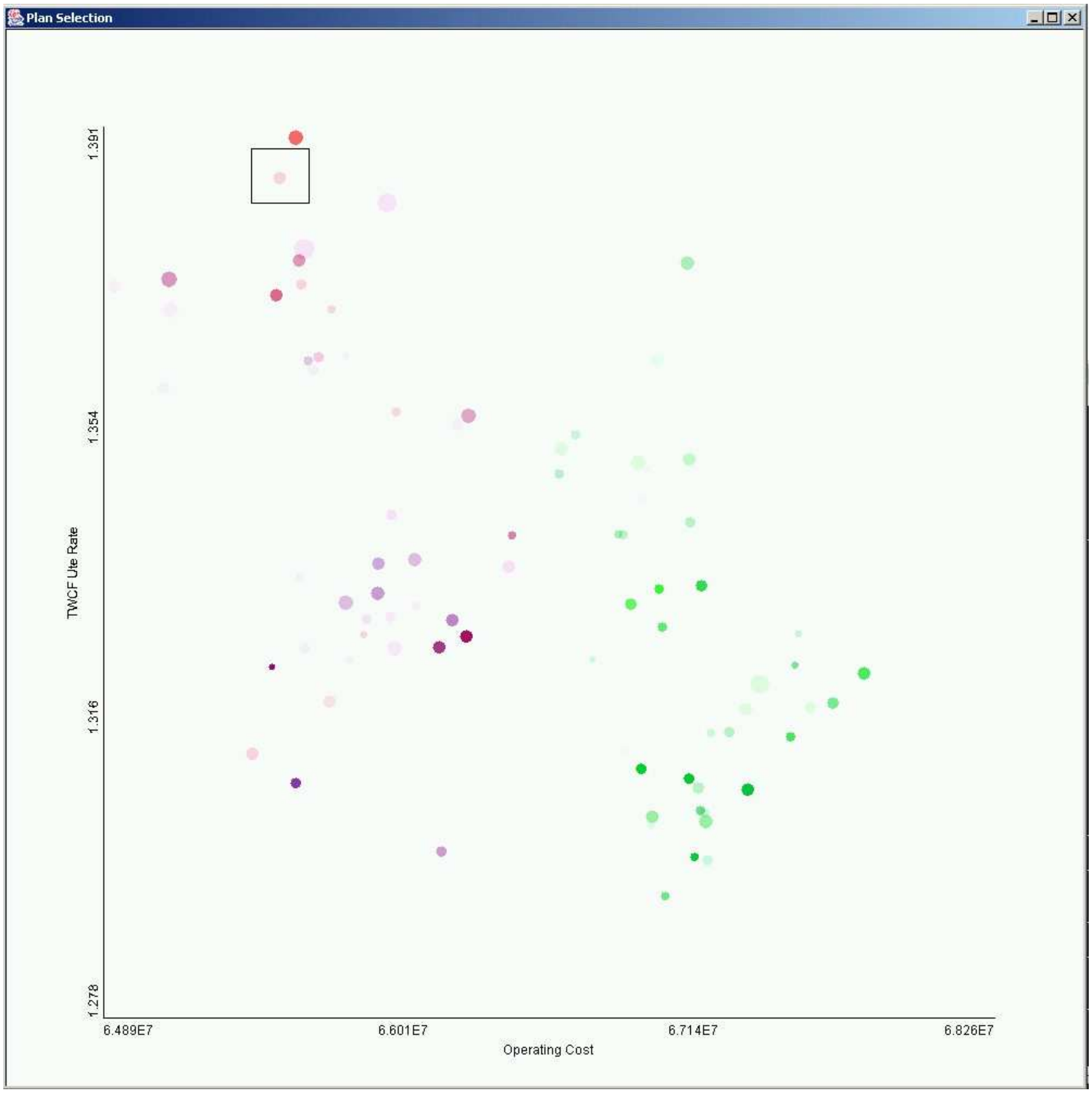


Figure 5-22: Second plan explored by user. The user does not consider this plan very deeply, because it has such a low probability, as indicated by its nearly transparent appearance

He then chooses the next closest, high probability plan (Figure 5-23). By inspecting the identifying information, he realizes that this plan has fewer C5-DOV and C17-CHS available. He is satisfied with the number of planes used in this plan, and after reviewing all the other details, he suggests its execution to his commanding officer (CO).

The CO is satisfied that the number of resources used by this plan will be available, but reminds the planner that the number of C17-CHS and C5-DOV planes available could vary. The planner goes back to see how variations in the number of these plane types affect the metric values. He selects the *one-dimensional score differential* tool, and he chooses to vary C17-CHS and C5-DOV individually around the baseline plan that he originally suggested to his CO (Figure 5-24). This tells him that the plan he suggested to his CO has the best TWCF Ute Rate when either C5-DOV or C17-CHS is perturbed; if the number of C17-CHS or C5-DOV increases or decreases, the TWCF Ute Rate will go down. However, if the number of C5-DOV is increased by one, the Operating Cost goes down. Likewise, if the number of C17-CHS is decreased by one, the cost goes down. These could be reasons for using a different set of resources. Neither graph exhibits particularly tight clustering, which indicates that the chosen plan is not very robust. Changes in the number of available C17-CHS or C5-DOV while planning will cause large changes in metric scores. This means that it will be expensive to switch plans in light of changes in resource availability. However, this is not a big concern, because the probability of the resources needed for the plan being available is high, so there is not a great need for robustness.

The CO reminds him that often C5-DOV and C17-CHS planes often become unavailable at the same time, so the planner goes back to look at those two simultaneous variations. He uses the *Two-Dimensonal Score Differential* tool. He allows those two values to vary around the baseline plan that he originally showed to the CO (Figure 5-25). These graphs tell him that it is possible to achieve higher TWCF Ute Rates with different sets of resources, but it will also cost more. However, for certain values of C17-CHS, such as 16, it is not possible to increase the TWCF Ute Rate, and decreasing the cost greatly decreases the TWCF Ute Rate (Figure 5-26(a)). On the

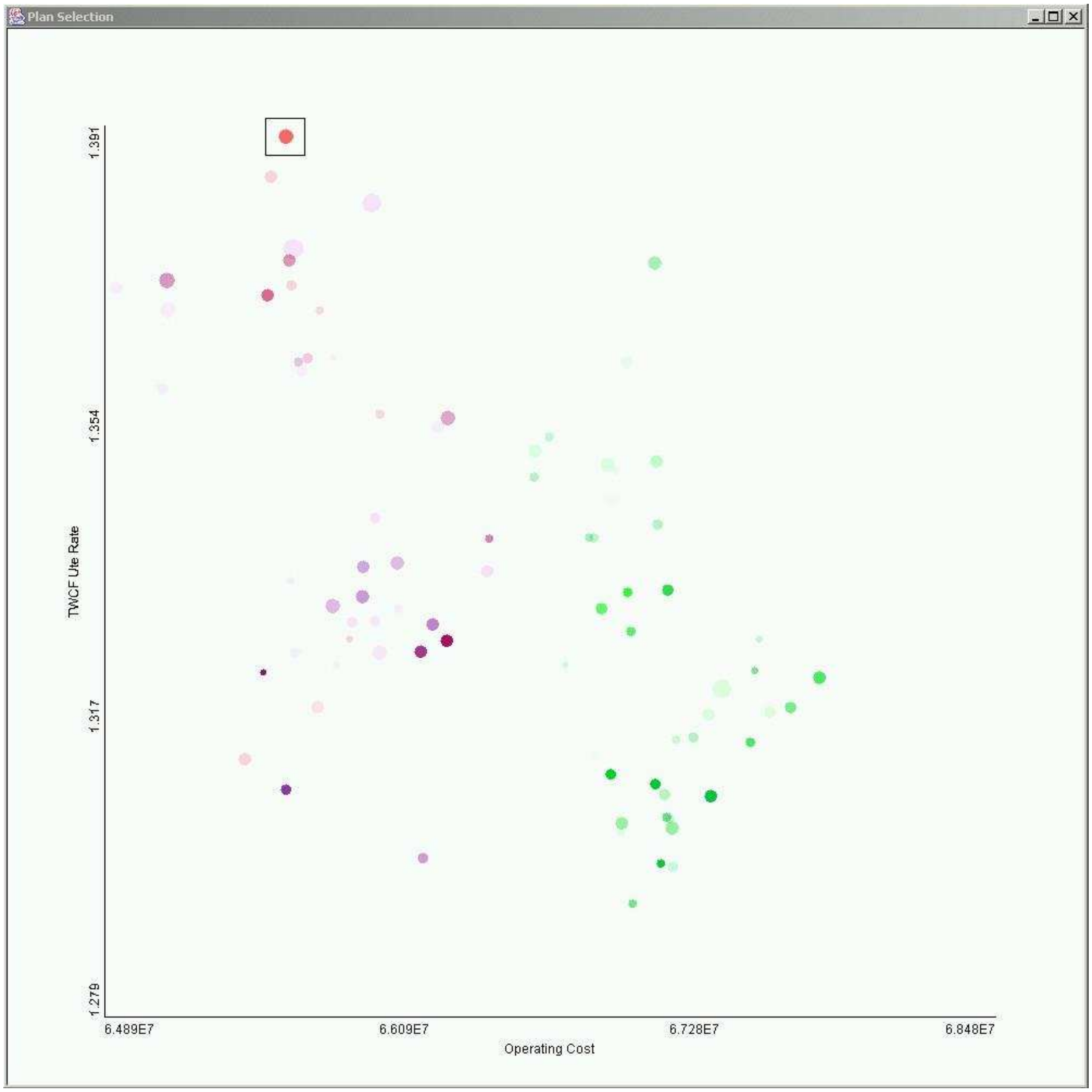


Figure 5-23: Third plan considered by user

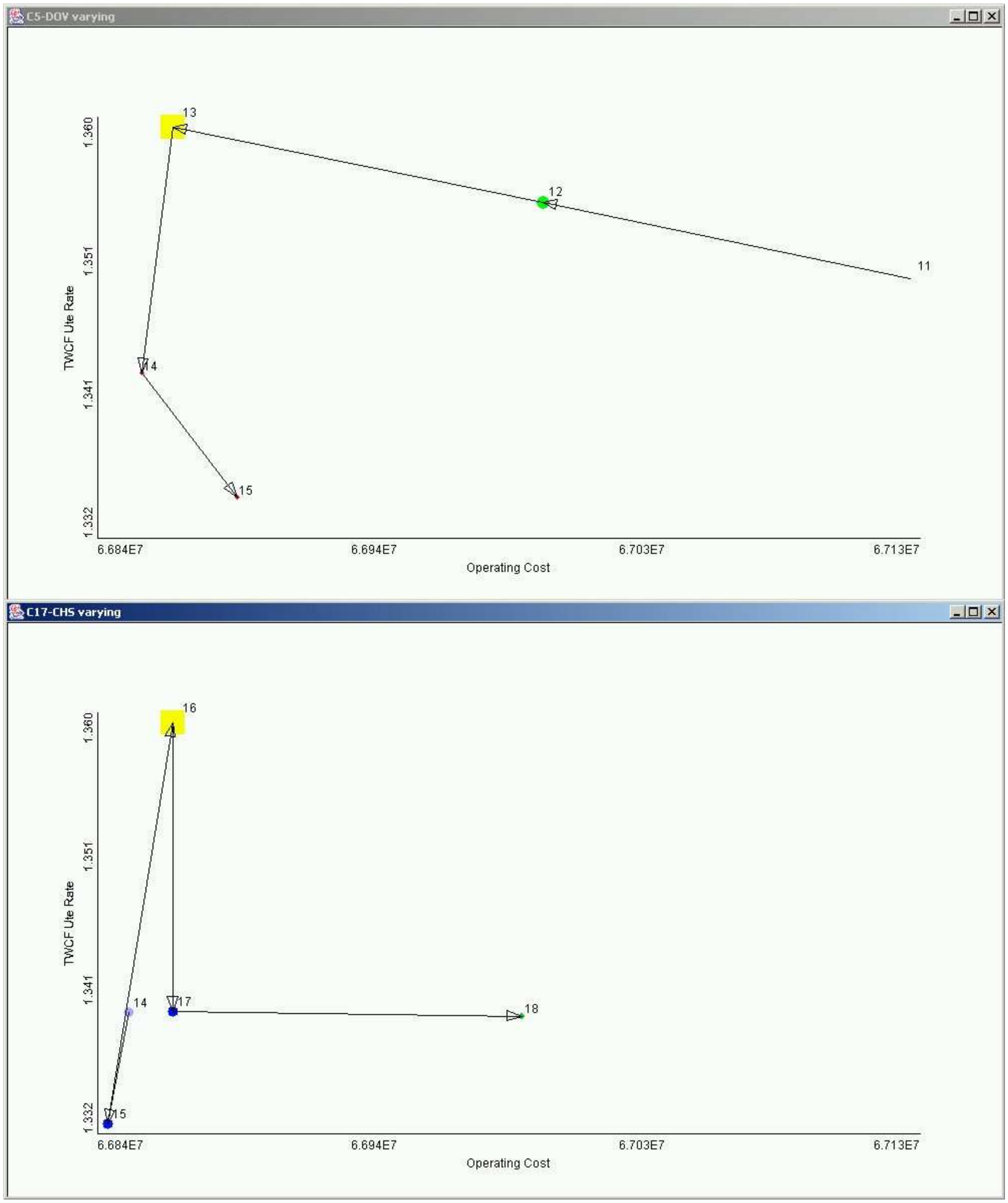


Figure 5-24: *One-Dimensional Score Differential* graphs viewed by user. The title bar of the top graph indicates variation in C5-DOV, while the one in the bottom graph indicates variation in C17-CHS. The upper left point in both graphs represents the same plan.

other hand, when there are 18 C17-CHS available, increasing the number of C5-DOV available by one greatly decreases the cost and even slightly increases the TWCF Ute Rate(Figure 5-26(b)). For this reason, this set of resources may be a better one to use. He then wonders about the plan's robustness. Although the dots on the graph do not appear to be tightly clustered, the range over which they fall is very small. The range of the x and y axes is only two percent of the value. Therefore, the differences in metric scores between the plans are small enough to consider the plan robust to changes during the planning process.

The planner reports his latest findings to the CO, who is finally satisfied. The CO accepts the plan for execution.

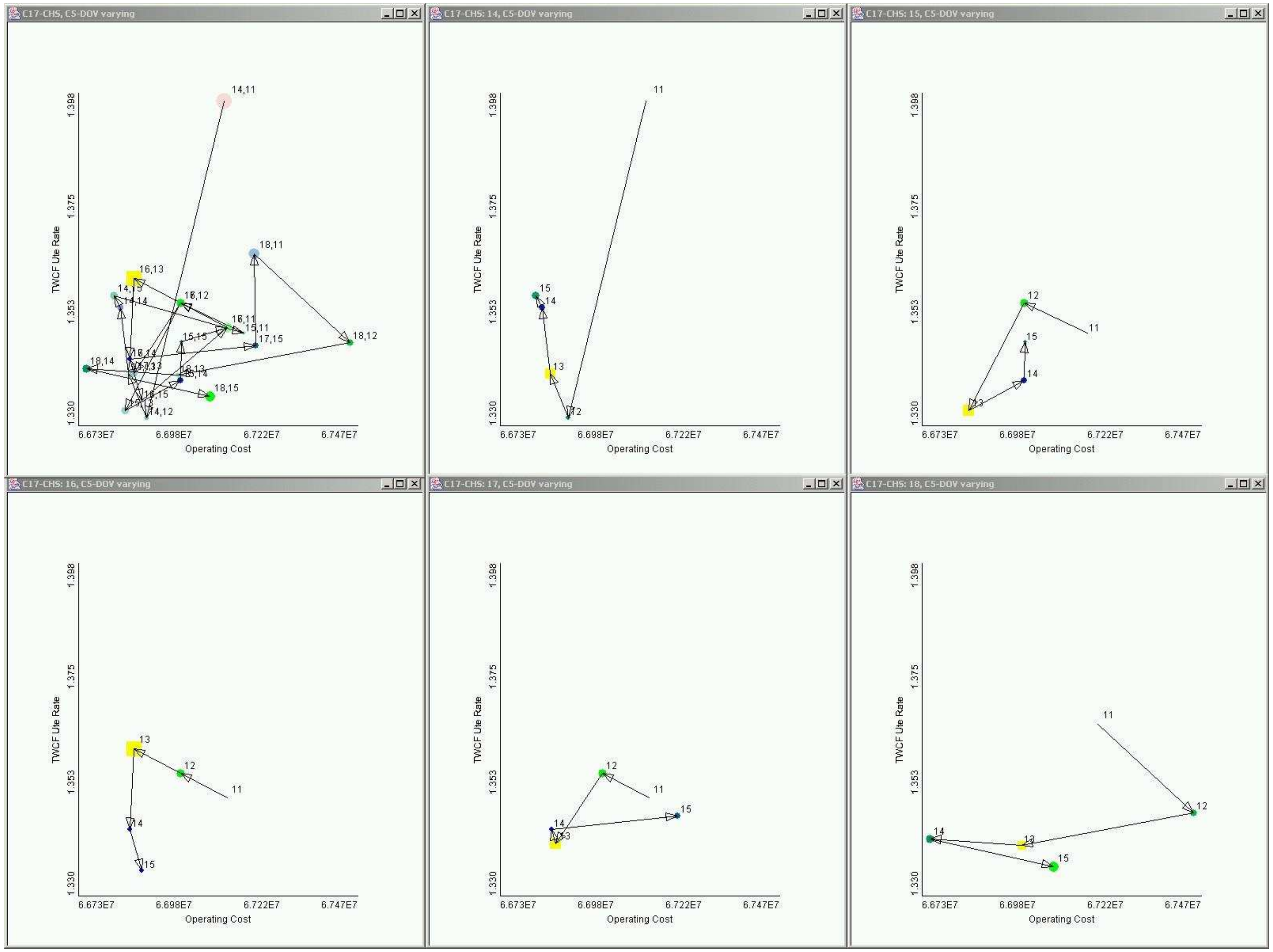
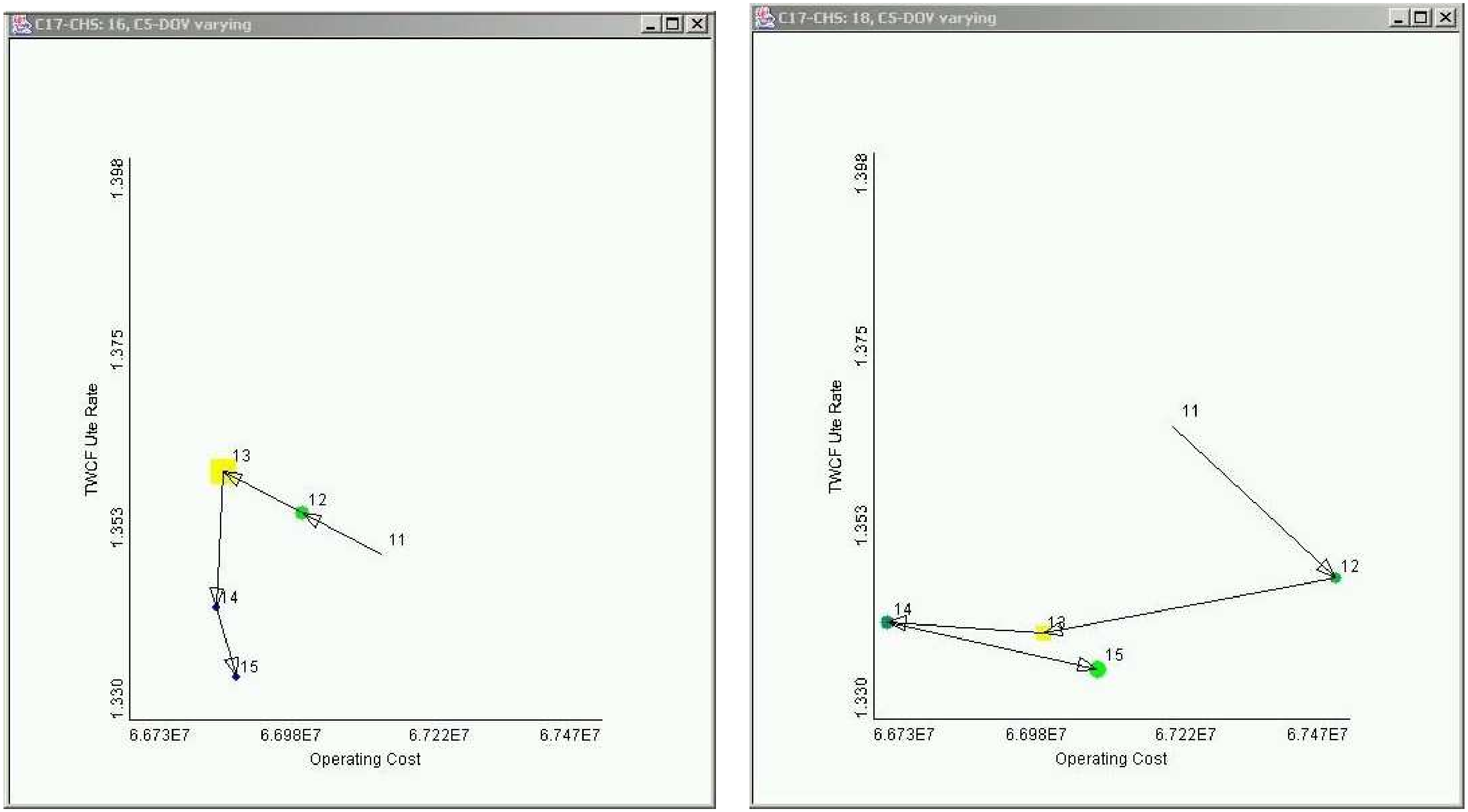


Figure 5-25: *Two-Dimensional Score Differential* graphs viewed by user. The top graph is the root graph that shows variations in both C5-DOV and C17-CHS simultaneously. The others hold the value of C17-CHS constant while varying the number of C5-DOV. For example, the bottom right graph holds C17-CHS at 18 while varying C5-DOV The baseline plan is surrounded by a yellow box in all graphs.



(a) (b)

Figure 5-26: Close up view of *Two-Dimensional Score Differential* graphs. (a) shows a graph with C17-CHS held at 16 while C5-DOV varies, and (b) shows a graph with C17-CHS held at 18 while C5-DOV varies.

Chapter 6

Implementation

6.1 Introduction

This chapter describes the implementation of ChRIS at the code level. The code implementing the system described in the previous chapters was written in Java^{TM}. Version 1.4 of the Java^{TM} Standard Development Kit was used to develop the system in a Windows 2000 Professional Edition environment. Version 1.4 of the Java^{TM}2 Runtime Environment is necessary to run ChRIS. The software necessary to compile and run the system can be found at www.javasoft.com.

Each of the three main Java^{TM} packages in the implementation will be described in the following sections by way of short descriptions of each of the classes within the package. Finally, instructions for compiling and running ChRIS are given. A UML Class Diagram describing the entire implementation is given in Figure 6-1[1]. The figure shows the relationship between the classes in the three main packages. There are two types of links between classes in the diagram. The lines of association links signify that one class uses an instance of or methods from the other. There is no difference between horizontal and vertical association links. The arrows of generalization links signify that the class pointed to by the arrow is the superclass of the class where the arrow started. Classes listed with italics are abstract classes.

---

[1] This figure was created using Visual Paradigm for UML Community Edition, which can be downloaded from http://www.visual-paradigm.com

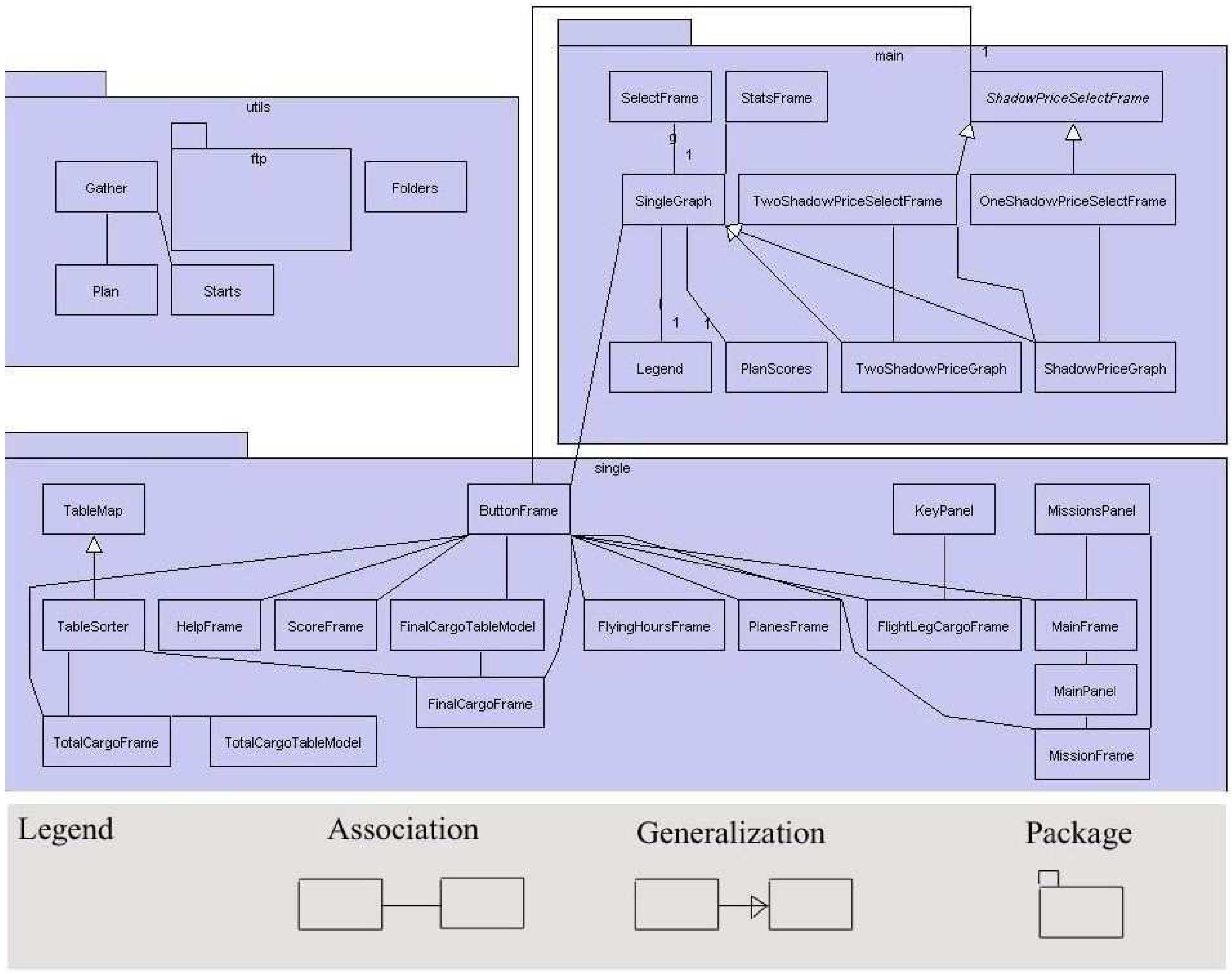


Figure 6-1: UML Class Diagram for entire implementation

6.2 Package single

This package contains the classes associated with displaying information about a single plan. The basic function of each class is described in Table 6.1.

Class Name	**Description**
ButtonFrame	This is the main class of package single. It implements the window that contains the *Options Selector* (Figure 4-14). It also contains an instance of each of the displays of information for a plan. In other words, it contains within it an instance of each of the other Frame classes in this package. These instances of the Frame classes are the ones that are actually displayed to the user. This was done to keep track of the displays for a plan and to ensure that more than one of a certain type of display is instantiated for a plan.
MainPanel	Draws graph of missions start times for MissionFrame and MainFrame (Figure 4-15). This class also handles the mouse input generated when the graph of *Mission Start Times* is visible.

Table 6.1: Classes in package single

Class Name	**Description**
MissionsPanel	Draws map of *individual missions* for MissionFrame and MainFrame (Figure 4-16). This class also handles the mouse input from clicking on flight legs to see the list of cargo on the plane at that time. Each flight leg is represented by a line segment on the graph, and each mouse click occurs at a point. To determine if the mouse was clicked on a flight leg, the distance between the click and all of the line segments is calculated, using an algorithm for calculating the distance between a point and a line segment[2]. If the distance is small enough, then the click occurred on that line segment, and the cargo for that flight leg is listed. Cargo is listed for the first line segment for which the distance is small enough, even if it is small enough for multiple lines (i.e. near intersections of lines).
MissionFrame	Implements the window in which the MainPanel and the MissionsPanel are displayed.
MainFrame	Implements a window that combines the functionality of the MissionFrame and the ButtonFrame. It does not have any functionality that is not provided by ButtonFrame or MissionsFrame. The buttons of ButtonFrame are along the left of the window, and the graphs of the MissionsFrame are displayed on the right side of the window.
KeyPanel	Draws the legend of colors to planes for FlightLegCargoFrame (Figure 4-17).

Table 6.1: Classes in package single

---

[2] taken from http://geometryalgorithms.com/Archive/algorithm_0102/algorithm_0102.htm

Class Name	**Description**
FlightLegCargoFrame	Implements smaller window that is displayed when viewing graphs of *individual missions*. It contains the KeyPanel on the left and the list of cargo on a flight leg on the right (Figure 4-17).
PlanesFrame	Implements the window that contains the *Available Planes* display (Figure 4-25).
TotalCargoFrame	Implements the window that contains the *Total Cargo* display (Figure 4-26).
TotalCargoTableModel	Subclass of the Java Class javax.swing.table.AbstractTableModel that tells TotalCargoFrame how to display its list of cargo.
FinalCargoFrame	Implements the window that contains the *Final Cargo* display (Figure 4-27).
FinalCargoTableModel	Subclass of the Java Class javax.swing.table.AbstractTableModel that tells FinalCargoFrame how to display its list of cargo.
TableMap	Subclass of the Java Class javax.swing.table.AbstractTableModel that tells a given TableModel, such as TotalCargoTableModel and FinalCargoTableModel how to display its data.
TableSorter	Subclass of TableMap that sorts the lists of data from a TableModel, such as TotalCargoTableModel and FinalCargoTableModel by columns.
ScoreFrame	Implements the window that contains the the *Plan Scores* window (Figure 4-28).
FlyingHoursFrame	Implements the window that contains the *Flying Hours* display (Figure 4-29).

Table 6.1: Classes in package single

Class Name	**Description**
HelpFrame	Implements the window that contains the the *Help* tool (Figure 4-30).

Table 6.1: Classes in package single

6.3 Package main

This package contains the classes associated with displaying information about multiple plans at once. The basic function of each class is described in Table 6.2.

Class Name	**Description**
SelectFrame	Implements the window that contains the the *metric selection tool* (Figure 4-4). It contains a SingleGraph, which is the graph created when the display button is pushed. Its dimensions are set to the metrics chosen on the SelectFrame.
SingleGraph	Implements the window containing the basic *Five-Dimensional Graph* (Figure 4-3). It also contains a inner class that implements the window containing the *legend* with dot color, size, and probability information (Figure 4-6). This class handles all the mouse input from selecting of points on the graph, selecting regions of the graph, and choosing options from the *Pop Up Menus*. An instance of SingleGraph contains a set of ButtonFrames, from package single. This serves as the main interface between the two packages.

Table 6.2: Classes in package main

Class Name	**Description**
OneShadowPriceSelectFrame	Subclass of ShadowPriceSelectFrame that implements the *One-Dimensional Score Differential* selection tool (Figure 4-21). An instance of OneShadowPriceSelectFrame is contained within each instance of ButtonFrame from package single. That instance of OneShadowPriceSelectFrame is used when the plan for which the ButtonFrame was created is to serve as the baseline for the single parameter variation.
TwoShadowPriceSelectFrame	Subclass of ShadowPriceSelectFrame that implements the *Two-Dimensional Score Differential* selection tool (Figure 4-23). An instance of TwoShadowPriceSelectFrame is contained within each instance of ButtonFrame from package single. That instance of TwoShadowPriceSelectFrame is used when the plan for which the ButtonFrame was created is to serve as the baseline for the dual parameter variation.
ShadowPriceSelectFrame	Abstract base class for the two different types of tools used to create score differential information, OneShadowPriceSelectFrame and TwoShadowPriceSelectFrame

Table 6.2: Classes in package main

Class Name	**Description**
ShadowPriceGraph	Subclass of SingleGraph that does the extra work for graphs where one parameter is varying (Figure 4-22). In this case, the additional work is filtering out all points not relevant to the current variation, ordering the remaining points in order of increasing value of the varying parameter, drawing the arrows between adjacent points in the ordering, and labeling of the points with the value of the varying parameter.
TwoShadowPriceGraph	Subclass of SingleGraph that does the extra work for graphs where two parameters are varying (Figure 4-24) Again, the additional work is filtering out all points not relevant to the current variation, ordering the remaining points in order of increasing value of the varying parameters, drawing the arrows between adjacent points in the ordering, and labeling of the points with the value of the two varying parameters.
StatsFrame	Implements the window containing the listing of average parameter values and metric scores (Figure 4-9).

Table 6.2: Classes in package main

Class Name	**Description**
PlanScores	Basic data structure storing parameter values and metric scores for individual plans. It contains fields for each of the parameters and metric scores, as well as an accessor and mutator function for each one. One PlanScores is created for each plan to be displayed, or for each run of the optimization.

Table 6.2: Classes in package main

6.4 Package util

Our standard procedure is to run the CRNPS on multiple sets of inputs, where each run creates a plan. A separate utility had been created that creates a directory for each plan's output files. The CRNPS input parameters are recorded in the name of the directory that holds its outputs.

The classes in the util package are utility classes that pre-process information from the optimization's output files. These classes do not use or depend upon the classes in the other two packages. The classes in this package move between the directories containing CRNPS output files and pre-process those output files for ChRIS, sometimes creating more files within those directories. The functions of the classes in this package are described in Table 6.3.

Class Name	**Description**
Plan	Calculates the metric scores for a given plan or run of the optimization. The processing method of this class returns a string listing of all the parameter values and metric scores for that plan. The parameter values are obtained by parsing the directory's name. The metrics for the plan are calculated after extracting information from the output files. The number of Missions Flown and Flying Hours is found from the listing of missions in *mission_statistics.out*. The number of aircraft is listed in *wrap_arcs.out*. The Operating Cost is listed in *CVF_Master.out*. The Hold Time is calculated from *Commodity_times.out*. The TWCF Ute Rate is calculated from *TWCF_ute_rates.out*, and the System Ute Rate is found from *Flt_leg_ute_rates.out*. The nature of these calculations is described in Section 6.4.1.
Starts	Pre-processes data from one plan, and creates files for later use by the different system tools. It creates a file listing the number of missions leaving each airport on each day. This file is used by MainPanel in package main. It also creates a file for each airport for each day that at least one mission starts from that airport. This file lists the flight paths of all the missions starting from that airport on that day. These files are then used by different instances of MissionPanel in package main. It also creates a file listing the Flying Hours for each type of aircraft in the plan, used by FlyingHoursFrame in package single. It creates a file listing the number of available planes for the plan, which is used by PlanesFrame in package single. The nature of these calculations is described in Section 6.4.2

Table 6.3: Classes in package util

```
Wrap around arc 16837 has 10 C5's from DOV
Wrap around arc 16873 has 13 C17's from CHS
Wrap around arc 16909 has 4 C17's from RMS
Wrap around arc 16945 has 3 C141's from WRI
Wrap around arc 16981 has 14 C130's from RMS
====================
TOTAL ACFT = 43
```

Figure 6-2: Structure of *wrap_arcs.out*: the keywords `TOTAL ACFT =` help indicate that there are 43 total Aircraft Used in the plan

Class Name	**Description**
Gather	Moves between directories of different runs of the optimization so that each plan can be processed. Specifically, it moves into the directory for a plan, creates an instance of Plan and Starts, and runs the processing methods of those two instances. It does this repeatedly until the outputs of all plans have been processed.
Folders	Transfers a copy of the optimization output between machines. This was used mainly to transfer optimization output from the machine where the optimization was actually run to the machine where the system development occurred.
ftp	Package downloaded[3] to facilitate transfer of optimization output between machines.

Table 6.3: Classes in package util

6.4.1 Processing of CRNPS Output Files by Plan Class

Some information is available directly in CRNPS output files. For example, the value of the Aircraft Used metric is listed in *wrap_arcs.out* (Figure 6-2), and the Operating Cost is listed in *CVF_Master.out*.

---

[3] from http://www.geocities.com/SiliconValley/Code/9129/javabean/ftpbean

Other metrics depend on the repetitive format of some output files, calculating the metric score involves noting the pertinent values from each repetition. For example, when calculating the number of Missions Flown and Flying Hours from *missions_in_solution.out*, the pertinent information is gathered from each mission. Figure 3-2 gives a sample mission from the file. Each block of this form in the file is counted to give the number of missions in the plan. The flying hours in a single mission are listed after the keywords `Flying Hours:`, and they are totaled for the plane types listed after the keyword `Aircraft:` (Figure 3-2). Similar methods are followed during the calculations of the average System and TWCF Utilization Rates and AMC Hold Times. Again, the relevant information is extracted from each instance of a repeated structure in the output file, where the relevant information is flagged by descriptive keywords.

6.4.2 Processing of Output Files by Starts Class

The number of missions leaving each airport on each day is also calculated from *missions_in_solution.out*. All missions leave from one of only four airports. The start location is the first Airport listed in each mission listing, and the start date is the next thing listed (Figure 3-2). The starts are tallied for ¡airport, date¿ pairs, and the tallies are written to *starts.out*. This information is then used by MainPanel to draw the graph of *Mission Start Times* for a plan.

The next task of Starts is to create a file listing the flight paths of all the missions leaving an airport on a given day. If no missions leave an airport on a given day, no file is created. The flight path of a mission is given in the mission listings of *missions_in_solution.out* (Figure 3-2). The lines in the sequence list the places visited on the flight path, the time of arrival at each location, and the time of departure from each location. All of the airports in the sequence are visited during the mission. This information is then translated and written to a file which can be used by MissionsPanel to draw the graph of *individual missions* (Figure 6-3). The first number in the first line of the file lists the number of missions leaving that airport on that day, which is consequently the number of missions whose flight legs are listed in the file. The other two numbers in that line are place holders. In this file, each mission is represented by an array of numbers with three numbers per line. The first row of each mission lists the mission number, the number of locations visited during the mission, and the plane type flying the mission. Then there are two lines for each location visited during the mission. In both lines, the airport is the first number, and the amount of cargo on the plane is the third number. The second number is the arrival time at that destination in the first line, and the departure time in the second line. There are mappings that translate airport names to numbers and plane types to numbers. Figure 6-3 gives a sample file for CHS on day 10, with comments added after selected lines. In all missions, the arrival and departure times are the same for the first location. This location is the home base of the aircraft flying the mission, and the mission begins with the first departure from the home base. Therefore, time of the previous arrival at the home base is irrelevant to the current mission, and it is set to be the time of the initial departure.

The cargo to be carried is also listed in the Loading section at the bottom of the mission information (Figure 3-2). The cargo assignments are indicated there as follows :

$$origin- > destination(\#id, \texttt{ALT} = ready - day, \texttt{tons} = tonnage)$$

A piece of cargo is on a flight leg if its destination has not been visited yet and if its origin has already been visited. If these conditions are true, then its weight is included in the total amount of cargo on the plane. When no cargo is carried on the plane, the cargo weight is listed as .2 tons. This is done for ease of drawing the graph of *individual missions*, where the amount of cargo carried on a flight determines its thickness on the graph. Listing 0.0 here instead of 0.2 would result in an invisible line of zero thickness, which is undesirable.

Notice in mission 310 of Figure 6-3, the plane departs airport 0 with no cargo and arrives at airport 1 with 55 tons of cargo. This signifies that the plane flew from airport 0 to airport 1 empty and was filled with cargo immediately upon arrival at

```
3 0 0  //3 missions leaving that day, place holder, place holder
310 4 1  //mission 310, 4 locations, plane type 1
0 10.0 0.2
0 10.0 0.2
1 10.0625 55.6784  //arrives airport 1 at 10.06 days with 55 tons of cargo
1 10.197917 55.6784  //departs airport 1 at 10.19 days with 55 tons of cargo
22 10.622916 0.2
22 11.445833 0.2
0 11.904167 0.2
0 12.0 0.2
640 4 1  //mission 640, 4 locations, plane type 1
0 10.0 0.2
0 10.0 0.2
17 10.045834 8.61289
17 10.18125 8.61289
19 10.552083 0.2
19 11.375 0.2
0 11.7875 0.2
0 12.0 0.2
670 4 1
0 10.0 0.2
0 10.0 0.2
17 10.045834 46.60764999999999
17 10.18125 46.60764999999999
20 10.514584 0.2
20 11.3375 0.2
0 11.691667 0.2
0 12.0 0.2
```

Figure 6-3: Sample file listing flight legs for missions leaving CHS on day 10, with comments after // inserted.

airport 1. Likewise, that the plane leaves airport 1 with approximately 55 tons of cargo and arrives at airport 22 with no cargo means that the plane flew from airport 1 to airport 22 full or cargo and was emptied immediately upon arrival. Therefore, the overall structure of this mission is as follows: plane leaves home airport 0, goes to 1, gets filled with cargo, takes that cargo to 22, gets emptied at 22, returns empty to home airport 0.

The Flying Hours are calculated in the same way as in class Plan, and they are written to a simple file consisting of four numbers, one for each type of plane. The number of available planes is listed in the file *input_problem_info.dat*, and it is obtained using the same method of searching for keywords in the file.

6.5 Compiling and Running ChRIS

When compiling and running the code, the classpath must point to the home directory of the code for the system. The code is organized into three sub-directories, one for each package. Compiling the file SelectFrame.java in the directory main will compile all of the necessary Java files for running the system. Using a standard command line compiler, such as javac, this can be done with the following command, run from the home directory:

```
javac -classpath . main/SelectFrame.java.
```

The *-classpath* switch sets the classpath to whatever follows it, which is the current directory in this case. To explicitly compile every file that will be needed, simply compile every Java file within the three packages. The following javac command run from the home directory will achieve this:

```
javac -classpath . main/*.java single/*.java util/*.java
```

Having compiled the code, the system can be run with the following command:

```
java -classpath . main.SelectFrame
```

This will display the *Metric Selection Tool*, which can be used to begin displaying information about all of the available plans.

Chapter 7

Conclusion

7.1 Summary and Contributions

The goal of this research is to reveal information needed to understand optimization solutions in order to make those optimization systems, specifically the Channel Route Network Planning System (CRNPS), more transparent. To that end, two major steps were taken, and the final result is the Channel Route Information System, or ChRIS, which was presented in this thesis.

First, we analyzed user needs for visibility into plans produced by an optimization, specifically the Channel Route Network Planning System (CRNPS). We determined that users need methods for quickly understanding the structure and details of the plan while not being overloaded with information. In order to satisfy this need, a set of tools were created that provide insight into the details of the plans created by the CRNPS. These tools include displays of includes *Mission Start Times*, *Individual Missions*, *Available Planes*, *Total Cargo*, *Final Cargo Locations*, *Plan Scores*, and *Flying Hours*. All of these tools reveal some information about the structure of the plan while not overwhelming the user with details.

Next, we analyzed user needs for information comparing multiple plans and determined the information needed to satisfy those needs. We decided that rather than initially compare the details of multiple plans, the user would want to compare general characteristics of many plans at once. To do this, we scored each plan using seven general metrics and compared the metric values for many plans graphically. We designed the *Five-Dimensional Graph* and its related tools to facilitate this comparison of metric scores. We also determined that users need insight into the differences between plan scores and the ramifications of changing plans. We investigated traditional shadow prices from optimization theory (Section 2.1) and determined that they are were insufficient because because they are not guaranteed to be valid over the entire range between two plans. Instead, score differentials were used, and the displays of one-dimensional and two-dimensional score differentials were created.

There are two main contributions of this work. The first is the actual system that the channel route network planners can use to choose and evaluate plans. This includes the methods for visualization of the details of a specific plan as well as for comparing the characteristics of multiple plans at once. The second contribution is the explanation of what information can be taken from each tool provided by the system. This explanation helps the user to get the most out of the system and make the best possible decisions.

7.2 Future Work

There are many areas that could be explored as extensions to this research.

Shadow prices could be used to indicate which further runs of the optimization should be done. Under the current mode of operation, the user sets the optimization to run a large number times and then begins to interact with the system using only that information. There is no concept in the software of doing more runs once the interaction has begun. As mentioned in Section 2.1, shadow prices indicate how the objective function changes with individual changes in the input variables. The user could use shadow prices to determine if he wanted to do more runs. For example, consider that while optimizing, the Operating Cost objective function is chosen. Also consider that user studies a plan that had 15 C5-DOV planes available, that the shadow price for available C5-DOV planes in this plan is negative, indicating that having more C5-DOV planes available would drive the Operating Cost down, but that no plans with more C5-DOV planes available have been generated. The user can then decide, based on the shadow price, to generate plans with more C5-DOV planes available. This functionality is not automatically supported, and could be added in the future.

Plan similarity or substitutability could be considered as a part of robustness. Currently, robustness is simply measured by proximity of points on the *Five-Dimensional Graphs*. Proximity indicates that as the available resources and, consequently, the plans change, the values of the plotted metrics do not change very much. Therefore, points that are close together on the graph are called robust. However, two plans with similar metric scores may be very different in structure, and it may be very hard to switch from one to the other during execution. On the other hand, if two plans are similar, then switching between them should be very easy. Currently, the only way to verify if plans are similar is to look at all of the information about the plans and make a judgment based on that. It would be better to find some way to quantify similarity of two plans and provide that information on the *Five-Dimensional Graph*. Then, plans represented by points that were close together on the graph and were similar could be called robust.

Statistical analysis of metric scores could be done. The system presented in this thesis focuses on visual means of displaying information about individual plans and comparing multiple plans. Perhaps a statistical analysis could discover relationships between the plans and their metric scores that are not easily seen in the visual displays of the system.

It would be beneficial to determine actual probabilities of resource availability. All of the discussion of probabilities in this thesis was based on the assumption that such probabilities could be found. However, a reliable source for those probabilities has not been identified. Were AMC to keep records of previous plans, the probabilities could be determined from that information. Until such data is available, it is difficult to determine how beneficial that information would be to a user.

Modifying this system according to human factors and GUI design principles would be helpful. The focus of this research was to identify the information that would be helpful to the user in understanding the structure of a plan and the ramifications of changing plans. We then tried to come up with novel ways of displaying that information. However, we did not attempt to design for usability or otherwise consider the human factors aspects of the displays.

Adding different modalities of interaction, such as speech or gesture would move the user away from the typical keyboard and mouse method of interaction. It would make the interaction between system and human more like it is between two humans, consequently speeding it up. Imagine that the user is looking at a *Five-Dimensional Graph*. He could simply say, “Show me all the plans with fewer than 350 missions,” and the system could then display only those plans on the graph. Then he could say “Show me the mission start times for that plan,” while pointing at a particular dot on the graph. The system could recognize which plan he is pointing at and show him the appropriate information.

Different display technologies could be explored. All development was done on a 1600x1200 pixel CRT display with 32 bit color. Perhaps different kinds of displays would provide greater visibility of the differences in darkness or color. They may also make layout of the many windows created by the system easier.

User studies of the system could be done to verify our judgments about what information is most beneficial to users in understanding the plans and to address usability issues. These user studies could verify that the users are finding the system useful. They will also indicate what is wrong with the system and how it can be fixed. The author has had preliminary conversations with two people familiar with the AMC planners and their planning process, but has not interacted with the AMC user directly.

There are also several minor changes to the displays that could be made. For example, some information, such as input parameter values, could be displayed in each window in order to identify the displayed plan. In the displays of individual missions, the overlap of lines could be eliminated to remove that confusion. In that same display, a legend relating line thickness to the weight of the cargo carried on that flight leg would be valuable, as would be a table of airport abbreviations. A display could be added that shows whether or not the cargo to be shipped around the network is actually making it to the correct destination at the right time. Different methods for viewing cargo information, instead of the list based methods used here, could also be designed.

There are three improvements that could be made the zooming behavior of the *Five-Dimensional Graph*. First, the user could be allowed to move around the graph while zoomed in, thus making visible parts of the graph that were once invisible. If this were possible, the user could just move to another region rather than zooming out and then zooming in again on the new region. Second, zooming out could occur in stages, rather than all at once, which is the case now. Third, the graph, when zoomed out, could display the boundaries of the regions that were previously zoomed in upon, in order to remind the user what he has looked at more closely.

To facilitate comparing average parameter values and metric scores for multiple regions of the graph, all subsequent average value displays could be made relative to the first one viewed. For example, if the first region viewed has an average number of missions of 285 and the second region has an average of 297, the first display would say 285, but the second one would say something like +12. Finally, the metric selection tool could be changed to use radio buttons (so that when a metric is chosen for a particular dimension, it is not allowed to be chosen for another), and to only enable the display button once metrics have been chosen for all four dimensions.

Appendix A

Glossary

Term	**Definition**	**Page**
ADA	Incirlik (Adana) Airfield, Turkey	34
AMC	Air Mobility Command. The office in charge of channel route network planning.	33
APOD	Aerial Port of Debarkation. The destination of a piece of cargo.	34
APOE	Aerial Port of Embarkation. The origin of a piece of cargo.	34
AVB	Aviano Air Base, Italy	34
BAH	Bahrain International, Bahrain	34
Basis	The set of linearly independent vectors that span the space	24
C5-DOV	A C5 Plane from Dover Air Force Base	35
C17-CHS	A C17 Plane from Charleston Air Force Base	35
C17-RMS	A C17 Plane from Ramstein Air Base	35
C130-RMS	A C130 Plane from Ramstein Air Base	35
C141-WRI	A C141 Plane from McGurie Air Force Base	35

Term	**Definition**	**Page**
CAI	Cairo East Air Base (Cairo International), Egypt	34
CDD	Crew Duty Day	36
CHS	Charleston Air Force Base, South Carolina	34
CO	Commanding Officer	112
Constraint	Function that limits the values a decision variable can take	23
CONUS	Continental United States	34
CRN	Channel Route Network. The network of airports through which cargo is shipped.	34
CRNPS	Channel Route Network Planning System. The optimization based-planning system that plans the shipment of cargo throughout the channel route network	38
Decision Variables	The variables over which the objective function is optimized	23
DOV	Dover Air Force Base, Delaware	34
EKJ	Prince Sultan (Al Kharj), Saudi Arabia	34
ESB	Esenboga (Ankara) Airfield, Turkey	34
Execution month	The month during which the plan is executed	37
FJR	Fujairah International Airfield, United Arab Emirates	34
Flight Leg	The components of a mission. A flight leg is the flight between two of the airports visited during a mission.	36
IGL	Cigli, Turkey	34

Term	**Definition**	**Page**
Initial Cut	First version of the plan created by the CRNPS and by the current manual planning process	37
JFACC	Joint Forces Air Component Commander	17
KEF	Keflavik Naval Air Station, Iceland	34
KWI	Kuwait International Airfield, Kuwait	34
LGS	Lajes (Air Base No. 4), Azores	34
Metrics	Functions used to score plans	46
Mission	A set of flight legs. A mission is responsible for transporting a set of cargo from origin to destination.	36
MHZ	Mildenhall Air Base, United Kingdom	34
NAP	Naples (Capodichino), Italy	34
NGU	Norfolk Naval Air Station, Virginia	34
NKW	Diego Garcia, British Indian Ocean Territory	34
Objective Function	The function to be optimized	23
OLB	Olbia (Costa Smeralda), Italy	34
Plan	A set of missions. The missions in a plan must transport all of the cargo to be shipped. A plan is created by the optimization planner.	37
Planning Period	The time during which the plan is created	37
PMOG	Parking Maximum on Ground. The maximum number of planes that can be parked at an airport at any time	34
RMS	Ramstein Air Base, Germany	34
RTA	Rota Naval Air Station, Spain	34

Term	**Definition**	**Page**
Shadow Prices	The amount by which the objective function changes for a unit change in an input	25
SIZ	Sigonella Naval Air Station, Italy	34
SKP	Skopje (Petrovec), Macedonia	34
THU	Thule Air Base, Greenland	34
TLV	Ben Gurion (Tel Aviv), Israel	34
TTH	Thumrait, Oman	34
TWCF	Transportation Working Capital Fund	40
TWCF Ute Rate	The percentage of aircraft capacity that is used on outbound CONUS flights	46
TZL	Tuzla, Bosnia Herzegovina	34
TZR	Taszar (Kaland), Hungary	34
WMOG	Working Maximum on Ground. The maximum number of aircraft that can be serviced at an airport at any time	34
WRI	McGuire Air Force Base, New Jersey	34

Bibliography

[1] Robert St. Amant. Planning and user interface affordances. In *Proceedings of the International Conference on Intelligent User Interfaces*, 1999.

[2] Andrew Armacost. Composite variable formulations for express shipment service network design. *Transportation Science*, 2002.

[3] Christopher D. Barth. Composite mission variable formulation for real-time mission planning. Master's thesis, Massachusetts Institute of Technology, 2001.

[4] Jacques Bertin. *Semiology of Graphics*. The University of Wisconsin Press, 1983.

[5] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[6] Jeff Johnson. *GUI Bloopers*. Morgan Kaufman Publishers, Inc., 2000.

[7] Christopher V. Jones. Visualization and optimization. *Interactive Transactions of OR/MS*, 1998.

[8] Alan M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, 1992.

[9] Christopher A. Nielsen. Large-scale network design using composite variables: An application to air mobility command's 30-day channel route network. Master's thesis, Massachusetts Institute of Technology, 2002.

[10] David Redmond-Pyle and Alan Moore. *Graphical User Interface Design and Evaluation (GUIDE)*. Prentice Hall, 1995.

[11] Arthur H. Robinson, Randall D. Sale, Joel L. Morrison, and Phillip C. Muehrcke. *Elements of Cartography.* John Wiley and Sons, fifth edition, 1984.

[12] Ben Schneiderman. *Designing the User Interface: Strategies for Effective Human-Compupter Interaction, Third Edition.* Addison Wesley Longman, Inc., 1998.

[13] Edward Tufte. *The Visual Display of Quantitative Information.* Graphics Press, 2001.

Index

Aerial Port of Debarkation (APOD), 34
Aerial Port of Embarkation (APOE), 34
Air Mobility Command (AMC), 33
airports, 34
Available Planes, 83, 108

barrelmaster, 37
basis, 24

cargo, 35
cargo bookie, 37
Channel Route Information System (ChRIS), 21, 43, 133
Channel Route Network (CRN), 33, 34
Channel Route Network Planning System (CRNPS), 38, 43, 91, 133
  outpupt, 40
class
  ButtonFrame, 119
  FinalCargoFrame, 121
  FinalCargoTableModel, 121
  FlightLegCargoFrame, 121
  FlyingHoursFrame, 121
  Folders, 126
  ftp, 127
  Gather, 126
  HelpFrame, 122
  KeyPanel, 120
  MainFrame, 120
  MainPanel, 119
  MissionFrame, 120
  MissionsPanel, 120
  OneShadowPriceSelectFrame, 123
  Plan, 125
  PlanesFrame, 121
  PlansScores, 124
  ScoreFrame, 121
  SelectFrame, 122
  ShadowPriceGraph, 123
  ShadowPriceSelectFrame, 123
  SingleGraph, 122
  Starts, 126
  StatsFrame, 124
  TableMap, 121
  TableSorter, 121
  TotalCargoFrame, 121
  TotalCargoTableModel, 121
  TwoShadowPriceGraph, 124
  TwoShadowPriceSelectFrame, 123
compiling, 130

composite variable formulation, 39
constraint, 23
cost, 23
Crew Duty Day (CDD), 36

decision variables, 23

execution month, 37

Final Cargo Locations, 85, 104
Five-Dimensional Graph, 46, 91, 93, 108
  Legend, 51
    Averages, 58, 93, 110
    Reset, 55, 98
    Scales, 53, 98
    Zoom, 55, 98
Flight Crews, 36
  augmented, 36
  basic, 36
flight leg, 36
Flying Hours, 88, 108

GUI Design, 28
Guide to User Interface Design and Evaluation (GUIDE), 29

Help, 89, 108
HSI Design, 28

implementation, 117
Individual Missions, 67, 104
  Legend, 67, 104
initial cut, 37

Java, 117

Metric Selection Tool, 50, 91, 93
metrics, 46
mission, 36
Mission Start Times, 66, 102
mixed integer program, 39
multi-dimensional tables, 28

objective function, 23
  possible, 40
optimization, 23
Options Selector, 63, 102
organic channel scheduler, 37

package
  main, 122
  single, 119
  util, 124
parallel coordinates, 28
Parking Maximum on Ground (PMOG), 34
plan, 37
Plan Scores, 87, 108
planes, 34
  types, 35
planning period, 37
Pop Up Menu, 59, 98
  id submenu, 59, 102
  options submenu, 59, 102
probability
  specification, 44

    use, 48
projection, 28

Retinal Variables, 28
running, 130

Score Differentials, 71
    One-Dimensional, 72, 111
    Two-Dimensional, 79, 112
shadow prices, 25, 134

Total Cargo, 84, 104

UML Class Diagram, 117

visualization techniques, 28

Working Maximum on Ground(WMOG), 34
worlds within worlds, 28

XpressMP, 39