Trust-Based Design of Human-Guided Algorithms

by

Joseph L. Thomer

B.S. Operations Research & Mathematics
United States Air Force Academy, 2005

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTERS OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

Signature of Author:

Sloan School of Management
Interdepartmental Program in Operations Research
17 May, 2007

Approved by:

Laura Major Forest
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by:

Cynthia Barnhart
Professor, Civil and Environmental Engineering
Co-Director, Operations Research Center
Thesis Advisor

Accepted by: ____

Dimitris Bertsimas
Boeing Professor of Operations Research
Co-Director, Operations Research Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 JUN 2007	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Trust-Based Design of Human-Guided Algorithms	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Sloan School of Management**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **AFIT/ENEL 2275 D. Street WPAFB, OH 45433**	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**229**	

[This Page Intentionally Left Blank]

Trust-Based Design of Human-Guided Algorithms

by

Joseph L. Thomer

Submitted to the Sloan School of Management
On 17 May 2007, in Partial Fulfillment of the Requirements
For the Degree of Master of Science in Operations Research

ABSTRACT

By combining the strengths of human and computers, Human Machine Collaborative Decision Making has been shown to generate higher quality solutions in less time than conventional computerized methods. In many cases, it is difficult to model continually changing problems and incorporate human objectives into the solution. Human-guided algorithms (HGAs) harness the power of sophisticated algorithms and computers to provide flexibility to the human decision maker to model correctly and dynamically the problem and steer the algorithm to solutions that match his/her objectives for the given problem. HGAs are designed to make the power of Operations Research accessible to problem domain experts and decision makers, and incorporate their expert knowledge into every solution.

In order to appropriately utilize algorithms during a planner's decision making, HGA operators must appropriately trust the HGA and the final solution. Through the use of trust-based design (TBD), it was hypothesized that users of the HGA will gain better insight into the solution process, improve their calibration of trust, and generate superior solutions. The application of TBD requires the consideration of algorithms, solution steering methods, and displays required to best match human and computer complimentary strengths and to generate solutions that can be appropriately trusted. Abstract hierarchy, Ecological Interface Design, and various trust models are used to ensure that the HGA operators' evaluation of trust can be correctly calibrated to all necessary HGA trust attributes.

A human-subject evaluation was used to test the effectiveness of the TBD design approach for HGAs. An HGA, including the appropriate controls and displays, was designed and developed using the described TBD approach. The participants were presented with the task of using the HGA to develop a routing plan for military aircraft to prosecute enemy targets. The results showed that TBD had a significant effect on trust, HGA performance, and in some cases the quality of final solutions. Another finding was that, HGA operators must be provided with additional trust related information to improve their understanding of the HGA, the solution process, and the final solution in order to calibrate properly their trust in the system.

Thesis Supervisor: Professor Cynthia Barnhart
Title: Professor of Civil and Environmental Engineering
Co-Director, Operations Research Center

Technical Supervisor: Laura Major Forest
Title: Human-System Collaboration Engineer
The Charles Stark Draper Laboratory, Inc.

[This Page Intentionally Left Blank]

Acknowledgements

17 May 2007

First, I must thank everyone who helped me with my thesis. Laura Forest provided the direction and support I needed to delve into a topic that was completely new to me. Alex Kahn reminds me how small the world is: We graduated from the same high school five years apart, sat next to each other in our combined middle and high school orchestra, and played in the pit band for the Pajama Game. I would still be coding in Java right now if Alex had not spent hours and hours of time transforming my ideas into code. The same goes for Moshe Shapiro, who spent hours turning power point slides into a workable GUI. I am very grateful of Prof. Cynthia Barnhart who had the dedication and patience to take on another student. Finally, I could not have completed this thesis without the two computer monitors provided by Jeff Cipolloni.

I must thank my lab mates who made struggling through homework sets almost worth the pain: John, Kiel, Keith, and Chris. I could not have done it alone – thanks guys.

I am thankful to my friends from SP who gave me a social life outside my tiny apartment: Joe, Brandon, Alison, Tom, and Andrea, who despite my teasing will make a good fighter pilot. I especially must thank Joe for his patience and friendship. The bar we built was amazing; too bad it's too big to fit outside the door.

Finally, I can say that my time at MIT was the best time of my life because of Mary Ashley. I unknowingly picked the farthest graduate school from Travis AFB, but every precious second I get to spend with her is worth it all.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research Project, Human-Machine Collaborative Decision Making.

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

Finally, as a member of the Air Force, I acknowledge that the views expressed in this thesis are mine and do not reflect official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

_____

Joseph L. Thomer
2nd Lieutenant, USAF

[This Page Intentionally Left Blank]

Table of Contents

Table of Figures

Table of Tables

Table of Equations

[This Page Intentionally Left Blank]

Chapter 1

Introduction

1.1 Problem Statement

As computers, automation, and decision-making software become prevalent, the issue of whether these devices can be trusted has become more important. With automation running nuclear power plants and military planners using decision-making software to plan air strikes, over-trusting these applications can have catastrophic consequences. Under-trusting these applications leads to these systems not being used to their full potential, which can result in wasted time, sub-optimal or infeasible solutions, or the application not being used and thus, effort wasted in developing the application. Determining the appropriate amount of trust one should have in an application, trust calibration, is essential to the proper use of automation and decision aids. Because an application that is not used correctly can be dangerous, it is the responsibility of the application designers to ensure their product is used correctly. Designers can build applications that help users to calibrate their trust and therefore improve the utility and effectiveness of their products.

Specifically, trust calibration is essential in Human Machine Collaborative Decision Making (HMCDM). HMCDM utilizes the strengths of both humans and computers to generate solutions for complex problems. A specific application of HMCDM, human-guided algorithms (HGAs), allows the user to control the algorithmic search for high quality solutions. With the use of HGAs, "better" solutions can be generated in less time than those created by either a human or computer alone. Improving calibration in HMCDM and HGAs can ensure the correct use of the application, improve decision making, and inform the user how much trust they should place in their final solution.

1.2 Motivation

Many others have identified the benefits of integrating human and machine decision making. Malasky (2005) has identified four classes of problems that can benefit from HMCDM and Parasuraman et al. have identified functions for complimentary HGA approaches. These approaches have been used to improve decision making in time-critical situations.

Malasky (2005) outlines four classes of problems that he believes would benefit from HMCDM. The first class is combinatorial problems, which require enormous computational time to find an optimal solution due to size of the required search space. However, a human can narrow down the search space and reduce the computational time required to solve the problem. The second class is visual problems, which benefit from pattern recognition abilities. An example is image classification or a neurologist deciding intensity and placement of beams of radiation that will irradiate a brain tumor without damaging surround tissue. The third class is computationally intensive problems that would require a human to evaluate the computer search progress and weigh the cost and benefits of continuing the search despite diminishing returns. The fourth class is heuristic heavy problems, which use rule-based heuristics to reach a satisficing solution in a reasonable amount of time. Allowing the human to select dynamically heuristics during the solution process might reduce computational time and improve solution quality. For HMCDM to be effective in these problems, HGAs need to harness the ability of humans to actively steer the algorithm's problem solving process.

Parasuraman, Sheridan, & Wickens (2000) have identified four automation dimensions based on their four-staged model of human information processing, which can be applied to human complimentary algorithms. The first dimension is information processing which involves sensing, filtering, organizing, or highlighting data. The second dimension, information analysis, predicts, integrates information, and augments human perception and cognition. Decision and action selection, the third dimension, involves selecting among decision alternatives. The fourth dimension, action implementation, includes the execution of the action. These automation dimensions are similar to the OODA loop. Developed by Col. John Boyd (1996), USAF, the decision process consisted of four components: Observe, Orient, Decide, and Act. Used for both military and business strategic planning, the manual OODA process can benefit from the integration of HMCDM and HGAs through the four classes of applied function for human complimentary algorithms. Using HGAs to find superior solutions in time-critical situations,

when compared to traditional methods, provides an advantage over the competition with more efficient solution techniques and more effective final decisions.

Currently, military commanders are skeptical of using automation in the decision-making process. Army officers, Schmitt and Klein (1999), comment in their Recognitional Planning Model, which they want the Army to adopt as a decision-making model, that the generation of a suitable course of action requires "the spark of creativity that generates the essence of a concept of operations" where there is "no procedural substitute for insight and expertise." Schmitt and Klein (1999) believe that the "black box activity" of automation will "generally result in sterile, unsatisfactory concepts." This belief highlights the requirement for research into how human-computer collaboration can remove the stigma of "black box" activity and intelligently incorporate valuable human input into complex solutions requiring the use of computers.

With the principles of HMCDM becoming increasingly relevant to problem solving and military and business applications, the issue of how to trust HMCDM applications must be addressed before society can begin to use this new technology successfully and safely. Understanding the strengths and weaknesses of any decision-making application can help a user make better-informed decisions and analyze the risks of implementing HMCDM solutions. Trust calibration will allow a user to improve their decision making, build appropriate mental models of HMCDM applications, and understand how to correctly control HGA technology to achieve faster, more reliable, better understood, and higher quality solutions than could be independently generated by either a human or computer.

1.3 Thesis Problem

In this thesis, methods for improving user trust are presented and the effects of incorporating the trust-based design (TBD) into HMCDM technology that uses an HGA to generate and display solutions to the mission planning of UAVs (Unmanned Autonomous Vehicles) are studied. The HMCDM software was originally developed to test the principles of HMCDM on the user's ability to quickly find and select quality solutions. It employs an HGA that allows the user to change the variable coefficients in the objective function. With the HGA, the user can steer a local neighborhood search out of local minimums and into a nearly global search. By searching the solution space, the user can gain a better understanding of the problem and find a solution that matches his/her unique human objectives.

Improved user control of the HGA and computer display will be designed around principles of TBD. Various TBD tools are evaluated and solutions generated with those tools are compared. Survey results from experimental participants will explain how the TBD tools affected their performance, trust, and decision making. In addition, trust calibration options will be discussed with the intent to improve trust calibration and decision making. Finally, the solutions generated with the HGA are compared to those generated with a global search algorithm.

1.4 Contributions

This thesis merges the design of automation with trust research and human-machine collaboration to improve the design of automation systems used to solve complex problems. The result was the theory of TBD for HGAs. Various trust models and theories applicable to automation are evaluated and applied to the development of TBD concepts. Controls and displays are developed with TBD to improve a vehicle routing HGA, which are also applicable to other domains. An experiment confirmed the utility of HGAs to solve complex problems and the requirement for decision makers to be provided with information that helps them evaluate trust in the HGA and the final solution. The experiment contributes to Operations Research by increasing the knowledge of human-machine collaboration, and developing new concepts for the development of algorithms that can be used by human operators to solve complex problems with novel solution techniques. This thesis is a small part of the required research necessary to remove the stigma of "black box" automation and decision-making tools.

1.5 Thesis Overview and Content

Chapter summaries are provided below.

1.5.1 Chapter 2: Automation and Trust Background Research

This chapter provides background research on the design of automation, human-machine collaborative techniques, HGAs, trust, decision making, and ecological interface design (EID). It provides a summary of the research considered for this thesis.

1.5.2 Chapter 3: Trust-Based Design of Human-Guided Algorithms

This chapter discusses the use of HGAs and applies trust theory to their design and use. An automation trust model, which is based on Lee and See's (2004) trust model, is presented to show how automation controls and displays affect human trust. TBD of controls, display, and trust information are explained.

1.5.3 Chapter 4: Human-Guided Algorithm for Vehicle Routing

This chapter explains the HGA used in the experiment conducted in this thesis. It describes the TBD tools and displays developed to improve the HGA and the rationale behind them. The idea of providing trust information to decision makers is applied to the HGA.

1.5.4 Chapter 5: Explanation of Trust-Based Design HGA Experiment

This chapter explains the TBD HGA experiment. It discusses the participants, purpose, design, training, hypotheses, procedures, and methods of data collection. The methods of data analysis are discussed.

1.5.5 Chapter 6: Analysis and Discussion

This chapter analyses and discusses all of the results of the experiment. Questionnaire results are explained and discussed. Additional observations that were recorded during the experiment are explained to provide further insight into the questionnaire results. A Discussion section combines the analyses and results into conclusions about TBD and HGAs.

1.5.6 Chapter 7: Summary and Future Work

The final chapter summarizes the thesis and discusses areas of future work with respect to applying TBD to other research areas, improving the HGA used in the experiment, and the further development of TBD and providing of trust information.

[This Page Intentionally Left Blank]

Chapter 2

Automation & Trust Background Research

2.1 Automation

This section summarizes the necessary considerations for the proper design of automation. Automation is the "execution by a machine agent of a function that was previously carried out by a human" (Parasuraman & Riley 1997). It is used to reduce human workload, decrease long-term cost, increase safety, or perform complex functions that humans are unable to perform. Automation is found anywhere from homes and workplaces, to cockpits, emergency rooms, and nuclear power plants. As technology and computers become more powerful, automation will continue to have an increasing impact on our daily lives. There are numerous ways in which automation can be employed and approaches for its proper design. Human Machine Collaborative Decision Making (HMCDM) is an application of automation that is used to solve complex problems.

2.1.1 Human Machine Collaborative Decision Making (HMCDM)

HMCDM is inter-disciplinary work done across fields that addresses the strengths of humans and computers to develop human-interactive decision systems. Current research objectives within HMCDM include:

- How can a human operator be effectively included in a complex decision-making system?
- Where can a human add the most value? Where can computer algorithms add the most value?
- What is the role of the automation? What is the role of the human operator?

- What degree/nature of human-machine collaboration do we assign in each point of the decision process?
- How can systems enable humans to augment or steer the computer algorithms based on qualitative and dynamic objectives?
- How can humans understand machine reasoning and come to trust the algorithms when using automation to augment their decision processes?

Currently, automated decision making faces the following problems: optimized plans do not always make sense to the operator, constraints are not easily defined by math, and operator preference is not incorporated into the search algorithms or heuristics. Heuristics are used to reduce the search space, improve computational time, and guide the search towards optimal or near-optimal solutions. However, developing heuristics that apply to all problem scenarios remains a challenge. HMCDM attempts to resolve these issues by allowing operators to implement, adapt, and augment heuristics dynamically depending on the scenario. The challenges of this approach include making the computer capable of incorporating operator suggestions and having the operator understand the machine's reasoning. HMCDM addressed human-computer collaboration at the beginning of the automation design process for the purpose of developing decision-making tools with increased dynamic flexibility, reduced human workload, and improved user acceptance and understanding.

2.1.2 Human Complementary Approach

HMCDM requires agents to agree on shared goals, communicate, plan, coordinate together, allocate responsibility, and possibly adapt and learn from each other. The two major approaches to HMCDM, defined by Terveen (1995), are human emulation and human complementary. Human emulation is similar to artificial intelligence and assumes that the human and computer have symmetric abilities. Human emulation adapts the computer to the human by focusing on the communication necessary for the computer to understand and think like a human. The human complementary approach assumes that humans and computers have asymmetric abilities. The unique capabilities of computers are used to complement humans. Responsibility is divided among the humans and computers to exploit the strengths and overcome the weaknesses of both agents. The goals of a particular automation can be accomplished by creating the proper

relationship between automation hardware, software, and user behavior (Sheridan 2002). This chapter focuses on the human complementary approach to automation.

2.1.3 HMCDM Design Considerations

Under supervisory control, defined by Sheridan (2002), human operators input or receive information from a computer that is controlling a process or environment. The human is "in the loop" and has the ability to control the automation to various degrees. A computer that removes the human from the loop is referred to as an autonomous controller. Adaptive automation is when human/machine control changes with time and the situation. Passing control back and forth between the human and computer is termed trading. Humans and computers also partake in sharing when they function together simultaneously.

There are a variety of techniques applied to determine the amount and type of human and computer interaction. These include task analysis and function allocation, the four dimensions of automation, evaluation of human bias and behavior, evaluative criteria, and theories of human use of automation.

2.1.3.1 Task Analysis and Function Allocation

A task analysis is used to break down the overall task into elements and determine how they relate in time, space, and function. For the overall task and subtasks to be completed successfully, the required information, decisions, control actions, and criteria for successful completion must be specified. Function allocation is then used to assign tasks to the human and computer. To aid in this process, P.M. Fitt (1951) created a list that states which tasks men or machines perform best. Due to changing technology, the Department of Defense (DOD) has since updated this list, which is shown in Table 1. While these lists are useful, function allocation should be dependent on what is best for the successful completion of the overall task, not just whether the function can be completed best by the human or machine. Depending upon the function allocation, the task analysis changes, therefore making the design process iterative.

HUMANS EXCEL IN	MACHINES EXCEL IN
Detection of certain forms of very low energy levels	Monitoring (both men and machines)
Sensitivity to an extremely wide variety of stimuli	Performing routine, repetitive, or very precise operations
Perceiving patterns and making generalizations about them	Responding very quickly to control signals
Ability to store large amounts of information for long periods, and recalling relevant facts at appropriate moments	Storing and recalling large amounts of information in short periods of time
Ability to exercise judgment where events cannot be completely predicted	Performing complex and rapid computation with high accuracy
Improvising and adopting flexible procedures	Sensitivity to stimuli beyond the range of human sensitivity (infrared, radio waves, etc)
Ability to react to unexpected low-probability events	Doing many different things at one time
Applying originality in solving problems: i.e., alternative solutions	Exerting large amounts of force smoothly and precisely
Ability to profit from experiences and alter course of action	Insensitivity to extraneous factors
Ability to perform fine manipulations, especially where misalignment appears unexpectedly	Ability to repeat operations very rapidly, continuously, and precisely the same way over a long period
Ability to continue to perform when overloaded	Operating in environments which are hostile to man or beyond human tolerance
Ability to reason inductively	Deductive processes

Table 1: **DOD List of Human and Machine Capabilities**
(U.S. DOD 1987)

Malasky (2005) expanded on the DOD list to create an expanded and updated list of human strengths and computer strengths.

Human Capability Strengths:
(Malasky 2005)

- Flexible/Adaptable: Humans can be flexible and adaptable in certain situations. A computer can only performed preprogrammed tasks.
- Creativity: Humans can think of original solutions and apply knowledge in unique ways.
- Visual Perception: Humans can quickly interpret and process visual information, while computers can easily display the information but not its meaning.
- Emotion: The human can abstractly consider objectives that are too complex for a computer.
- Learning from Experience: Humans can use previous experience to select or discard options that will or will not work for a scenario.
- Complex Communication: Computer communication is limited to printouts, screen displays, and sound. Examples of complex human communication include body language and tone of voice.
- Conceptualization: Humans can formulate ideas mentally and develop fuzzy and imprecise ideas into something more specific and precise.
- Symbolic or Spatial Reasoning: Humans can mentally manipulate abstract symbols and use them to make logical decisions.
- Intuition: Previous experience gives humans instinct and intuition that can reduce the time required to make a decision.
- Pattern Recognition: Humans can classify and describe patterns and use them to predict future behavior.
- Hedging Against Uncertainty: Humans can anticipate possible future states, predict what can go wrong, and try to prevent it from occurring. A computer cannot model everything that could go wrong but the human can develop solutions that account for uncertainty.
- Narrowing Search Space: Mitsubishi Electric Research Laboratories (MERL) have shown that humans can narrow the computer search space to allow computers to find optimal solutions faster.

- Management of Computational Effort: MERL has also shown than humans can effectively manage computer computational time by stopping the computer's search process at the point where the cost of continuing the search is not worth the improvement in the solution.
- Strategic Assessment: Humans can consider many strategies during problem solving. The computer is limited to its preprogrammed capabilities.
- Understanding the "Big Picture": A human can understand a solution's impact outside of the system for which it was developed.

Overall, having a human in the loop allows for well-informed decisions despite the absence of correct and complete information (Ruff, Narayanan, & Draper 2002).

Computer Capability Strengths:
(Malasky 2005)

- Displaying Information: A computer can quickly display data and graphical information in a flexible format that can easily be interpreted by a human.
- Data Management: Computers can store and retrieve a tremendous amount of data.
- Simple Repetitive Decisions: Computers can be programmed to perform simple repetitive tasks without making mistakes. Humans will not always perform tasks correctly or in the same manner.
- Performing Calculations: Computers are faster and more accurate at mathematical calculations.
- Combinatorial Problems: Combinatorial problems are large problems that have a large number of variables. Computers are better at solving these problems because humans cannot fully understand the problem due to the many variables and possible solutions.
- Continuous Availability: Computers can be used at any time; they do not adhere to work schedules or tire after long periods of work.
- Fast Computational Parallel Reasoning: Computers can perform many tasks at the same time. Humans perform most tasks in a serial manner.
- Speed: Computers can act much faster than humans can, an important strength for time critical applications.

- Accuracy: All humans make minor mistakes. The computer is always accurate if it is programmed correctly.
- Predictability: Computers use a precise binary '1' and '0' or 'yes' and 'no' language. There is no 'maybe'. Humans can be unpredictable.
- Low Cost: Computers can be very cost-effective in the long run, despite high cost initial investments.
- Risky Situations: Computers can replace humans in risky situations. The U.S. military currently uses unmanned aerial vehicles (UAVs) on reconnaissance and attack missions, thus avoiding the risk of endangering pilots.

In addition to determining which tasks to automate, the amount of and frequency of human interaction with the automation must be decided. Parasuraman, Sheridan, and Wickens' (2000) Levels of Automation of Decision and Action Selection scale, shown in Figure 1, provides ten different degrees of automation, ranging from complete human control to complete machine control. Taking into account techniques such as trading, sharing, and adaptive automation, there are numerous options for a designer to consider.

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. Informs the human only if it, the computer, decides to
	8. Informs the human only if asked, or
	7. Executes automatically, then necessarily informs the human, and
	6. Allows the human a restricted time to veto before automatic execution, or
	5. Executes that suggestion if the human approves, or
	4. Suggests one alternative
	3. Narrows the selection down to a few, or
	2. The computer offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all decisions and actions.

Figure 1: Levels of Automation of Decision and Action Selection 10-Point Scale

(Parasuraman, Sheridan, Wickens, 2000)

2.1.3.2 Four Dimensions of Automation

Parasuraman, Sheridan, and Wickens (2000) provide a framework for defining automation dimensions based on human information processing. They have broken human information processing down into four stages:

1. Acquisition and registration of multiple sources of information.
2. Conscious perception: manipulation of retrieved information.
3. Decision reached by cognitive processing.
4. Implementation of response or action consistent with decision choice.

These stages are not always serial; they can be overlapping and cyclical. The four dimensions of automation are based on the four stages of information processing:

1. Information Acquisition: sensing and registration of input data, highlighting, filtering, and organizing information.
2. Information Analysis: predicting based on data, integrating and managing information, augmenting human operator perception and cognition.
3. Decision and Action Selection: selecting among decision alternatives, requires information about tradeoffs between cost and value.
4. Action Implementation: executing the selected action, automatically executing necessary sub-tasks.

Parasuraman, Sheridan, and Wickens' stages of human information processing closely match another model for human decision making, the OODA Loop.

The OODA loop stands for Observe, Orient, Decide, and Act. Col. John Boyd, USAF, developed the decision model to aid the effectiveness of fighter pilot training and performance (see Figure 2). Boyd believed that understanding an opponent's decision cycle would provide an advantage. Currently the OODA Loop is used in military and business strategy.

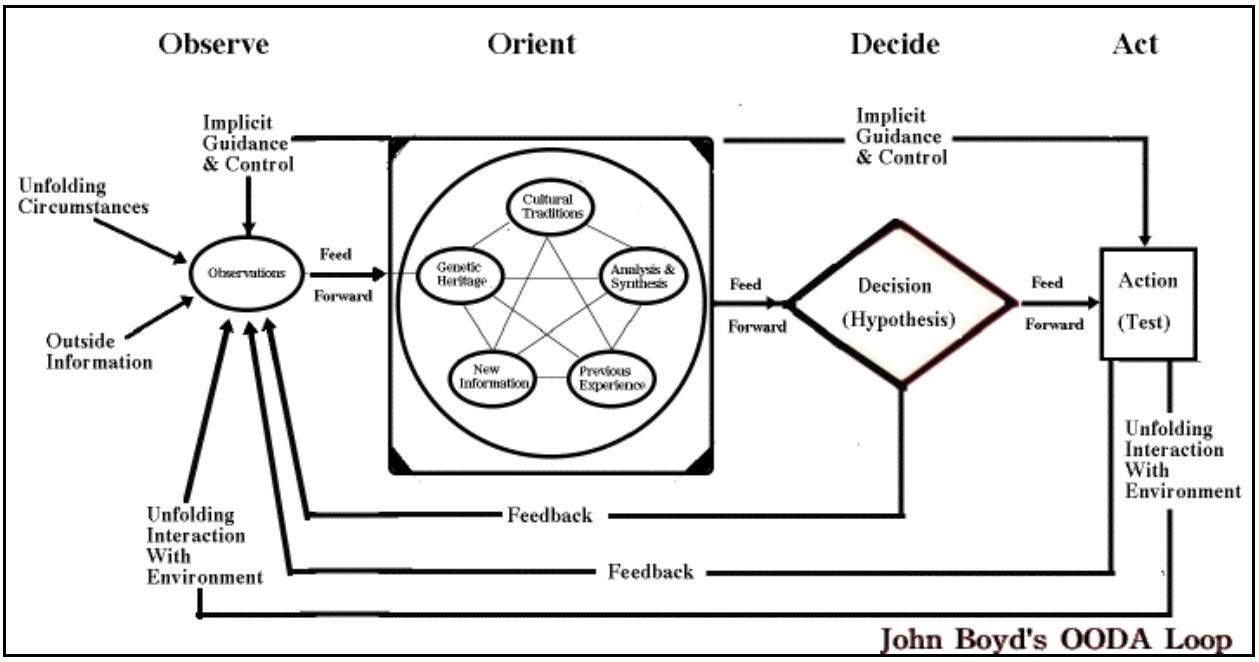


Figure 2: OODA Loop
(qtd. in Moran 2006)

2.1.3.3 Human Bias and Behavior

The theories on human behavior can also be considered when designing automation. Humans can face many biases including decision-making experience, willingness to accept risk, relative weighing of cost to self, and the tendency to decide on impulse (Sheridan 2000). Sheridan (2000) has composed a list of ways that humans deviate from rational norms:

- Decision makers do not give as much weight to outcomes as Bayes' rule would indicate.
- Humans tend to neglect base rates. They overweigh recent evidence and neglect previous evidence.
- Humans ignore the reliability of evidence.
- Humans tend to overestimate the probability of interdependent events and underestimate the probability of independent events.
- Humans seek out confirming evidence and disregard disconfirming evidence.
- They are overconfident in their predictions.
- They tend to infer illusory causal relationships.

- They tend to recall having greater confidence in an outcome's occurrence or nonoccurrence than they actually had before the fact. Hindsight bias: "I knew this would happen."

Muir (1994) points out additional biases:

- Humans overestimate predictability based on overestimating the representativeness of a small sample.
- Humans overestimate dispositional factors and underestimate the role of environmental factors attributing to the cause of one's behavior, which can result in humans underestimating predictability and dependability.
- Faith can be limited by a human's lack of understanding in the system, but can be increased by "blind faith."
- Humans have bias against automation because the consequences of automation failure may be high.
- The fact that designers want a human to monitor a system may imply that the system cannot be trusted.

2.1.3.4 Evaluative Criteria

In addition to biases, Sheridan, Wickens, and Parasuraman (2000) have proposed primary and secondary evaluative criteria that can be used to help in determining function allocation and the level of automation.

The primary evaluative criteria include mental workload, situational awareness, complacency, and skill degradation. Correctly designed automation matches the level of mental workload required by the user to an appropriate level, which keeps the user mentally engaged in the task, but also prevents the user from becoming overwhelmed. When automation is designed incorrectly, situation awareness, complacency, and skill degradation lead to safety issues during automation failure. These issues usually occur when the human is left "out of the loop" by higher levels of automation. Sheridan (2002) defines situational awareness as "the perception of the elements of the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the future." While perfect situational awareness

does not guarantee a user will always make correct decisions, inadequate situational awareness can result in the user making wrong decisions. Constant monitoring of a system can result in decreased situational awareness as the monitoring becomes subconscious to the user (Sheridan 2002). This is known as complacency, or overtrust in the system. The user is therefore less likely to notice changes in the system or remember details. Skill degradation occurs when automation performs a task that a human is trained to do, and therefore the user does not perform the task. This causes the human to lose the required skills necessary to perform the task, which may be needed during an automation failure. The secondary evaluative criteria include automation reliability and the cost of decision/action outcomes. Automation reliability is necessary for the primary evaluative criteria to hold true. Reliability has a direct effect on human trust in the automation, which changes how the operator uses the automation. The cost of decision/action outcomes is dependent upon the risk of automation failure. Low risk applications are prime candidates for higher levels of automation because the cost of automation failure is small. High-risk applications should be considered for lower levels of automation due to the large cost of automation failure. However, high-risk time-critical scenarios can be considered for high automation when the human does not have time to make a decision (Parasuraman, Sheridan, & Wickens 2000).

Parasuraman, Sheridan, and Wickens (2000) conclude, "the burden of proof should then be on the designer to show that their design will not lead to the problems of loss of situational awareness, complacency, and skill loss." In addition to these evaluative criteria, Malasky (2005) proposes additional evaluative criteria relevant to the use of automation in Human Machine Collaborative Decision Making (HMCDM) and optimization:

- Human Tendency for Boredom: If the human is "out of the loop" for a long period of time boredom may set in, resulting in complacency.
- Trust: The amount of trust the user places in the system affects how the operator interacts with the system. A user that is actively involved in the decision-making process is more likely to trust the final solution generated by the automation.
- Skill Set Requirement: Automation changes the operator skills necessary to complete a task; some skills become obsolete while new skills are needed to run the automation.

- Human Team Dynamics: When multiple users are involved in automation, team dynamics play a role in automation usage. Automation can also benefit from a team of users, rather than a single user.
- Human Operator Prior Experience: Previous experience can prevent the operator from adapting to the new skills required for automation. However, previous experience could also benefit a user.
- Recovery from System Failure: Automation can introduce new classes of human errors. For the operator to recover from system failure, the user must understand the decisions made internally by the automation. A lack of situational awareness can prevent successful recovery.
- Decision Interactions: Choosing levels of interaction for an automated task should be examined by its affects on other tasks and the overall system.
- Responsibility: The amount of responsibility an operator faces for a decision will determine the operator's willingness to accept an automated solution. Providing the user with information about the automation's decision making will give the user more confidence in the solution.

2.1.3.5 Automation Use

Parasuraman and Riley (1997) define four ways in which automation can be employed. Automation use is defined as the "voluntary activation or disengagement of automation by human users." Disuse is an under-reliance of automation resulting in it not being used to its full potential. Misuse is over-reliance on automation. Abuse is the inappropriate allocation of automation by designers and managers. How an operator decides between manual and automatic control of a system is a necessary consideration of the automation design process. Poorly designed automation, which has the potential to reduce human workload, may result in increased human workload or the system not being used at all. Automation that does not accurately reveal the constraints of its correct use may result in abuse, a condition that can be very dangerous inside a cockpit or nuclear power plant. Automation that helps the operator correctly determine when to implement automatic control will likely result in its successful use. Designers and operators ideally allocate automation to tasks while avoiding misuse, disuse, and abuse.

Misuse of automation can lead to errors caused by automation bias. Moser and Skitka (1996) define automation bias as "the tendency to use automaton cues as a heuristic replacement for vigilant information seeking and processing, resulting in errors when decision makers fail to notice problems because an automation aid fails to detect them (an omission error) or when people inappropriately follow an automated decision aid directive or announcement (commission error)." Heuristics are decision-making rules that can quickly deal with a lot of information and must be accurate most of the time. By balancing speed and simplicity with accuracy and reliability, heuristics are an alternative to cognitively demanding analytical decision making. Since humans are likely to take the cognitive path of least resistance, heuristics replace more demanding analysis and decision making. Heuristics are not always perfect and over-reliance on them and the automation that implements them can lead to errors. Systems that provide little system state feedback combined with an operator's incomplete or inaccurate mental model can be the largest obstacle to preventing errors. Misuse can also result in complacency, reduction in situation awareness, and skill degradation. Increasing system transparency without overloading the user with information can help prevent misuse. Transparency is defined as "how well the system interface reveals relevant information to help the operator estimate the state of the automation" (Gao & Lee 2005).

Abuse refers to the limits placed on operators to allocate manual or automatic control by their managers or automation designers. Just because a task can be automated does not mean that it should. Automation can result in negative consequences to the user and overall system, such as increased human monitoring and workload. In addition, the operator must be provided with enough information to decide between automatic or manual control and operate the system as designed.

2.2 Human-Guided Algorithm Systems

Algorithms that can be steered through human input are classified as human-guided algorithms (HGA). Shahroudi (1997) explains four methods of solution steering that can be employed by human operators to steer algorithms. Past work has employed human-guided algorithm concepts to develop research based systems that can be steered by human users. This work includes a Human Guided Search (HuGS) developed at the Mitsubishi Electric Research Laboratories (MERL) and a mission planning system developed at the Draper Laboratory. Experimental

results from both of these systems demonstrate that human controlled solution steering generates higher quality solutions in less time than can be generated by the human or computer alone.

2.2.1 Solution Steering

One particularly useful application of HMCDM is to provide a decision maker with a variety of methods to generate solutions to complex problems. Intelligently searching the solution space allows the user to seek out and analyze solutions with a uniquely human objective function. Humans can add flexibility to the solution process by finding solutions that incorporate objectives and constraints that are not programmed into the algorithm. Solution steering allows the user to modify variables, constraints, objectives, and control/tuning parameters of the numerical search strategy (Shahroudi 1997). The requirements for solution steering include a graphical display of information to the user, controls for the user to steer the solution, and algorithms and solvers that can be steered dynamically.

The four modes of control for solution steering are direct control on the search progress, direct control on the search problem, indirect control on the search problem, and direct control on the numerical search strategy (Shahroudi 1997).

2.2.1.1 Direct Control on the Search Progress

Direct control on the search progress allows the user to guide the location and direction of the search continually. Capitalizing on the user's ability to remember or imagine shortcuts can possibly accelerate the solution process. If the numerical search strategy is going in the wrong direction, the user can point it closer to the optimum. Local search algorithms can also become more global through this strategy.

2.2.1.2 Direct Control on the Search Problem

Direct control of the search problem allows the user to refine the problem definition continually, explore new solution options, and robustly plan for alternative scenarios. The user controls the parameters and coefficients used to define the objectives and constraints of the problem. By continually exploring the problem definition, it is possible for the user to explore the solution space, gain a greater understanding of the problem, and arrive at an improved problem definition.

2.2.1.3 Indirect Control on the Search Problem

Indirect control on the search problem is the inverse of direct control. Instead of changing the problem definition to see how the solution changes, the user mentally constructs what type of problem specification is needed to arrive at a certain solution. The benefits of indirect and direct control on the search problem are similar. The combination of these two methods makes it possible to explore problem and solution alternatives.

2.2.1.4 Direct Control on the Numerical Search Strategy

Under direct control on the numerical search strategy, the user selects among various search algorithms: local, global, probabilistic, deterministic, greedy, various heuristics, etc. The user can also tune the algorithm if appropriate.

2.2.2 Human Guided Search (HuGS)

MERL has focused on solution steering through direct control on the search progress and the numerical search strategy. MERL contends that allowing users to steer an algorithm based on user knowledge and preferences is superior to unguided algorithms that model oversimplified formulations of real-world problems (Klau et al. 2002 "Tabu", Klau et al. 2002 "HuGS", Rabiej 2000, Scott et al. 2002). Humans can take advantage of their superior ability of visual perception, learning, and strategic thinking. It is often impossible to specify in advance all of the constraints and selection criteria for all possible scenarios of the problem. Involving the human in the solution process allows scenario specific constraints to be incorporated. In addition, it is easier for operators to trust, justify, and modify solutions that they help generate, compared to automatically generated solution.

HuGS (Klau et al. 2002) is interactive software that allows a human operator to steer the internal search algorithm. A visual display allows the user to constrain the algorithm's exploration of the solution space. HuGS allows two primary search strategies:

1. User focuses the internal search algorithm on a restricted portion of the solution space; or
2. User modifies a solution to pull the algorithm into a new part of the solution space.

These strategies can be accomplished by the following three capabilities to:

1. Manually alter a current solution;
2. Invoke, monitor, or halt a focused search for an improved solution; and
3. Revert to previous or pre-computed solutions.

A visual user interface is necessary for the human to interact with the algorithm. The visualization is problem dependent, but must contain three important capabilities, namely it must:

1. Allow the user to select manual moves and then report them to the search algorithm;
2. Display current solution and user defined priorities (mobilities) when prompted; and
3. Revert to a previous or pre-computed solution when prompted.

The HuGS visualization used a tabletop display that allowed the user to implement the above capabilities. Displaying the changes made by the algorithm to a solution was found to be another useful capability. The user could also select from several display options that range from allowing the automation to run in the background while displaying the best solution, to having the user step through every solution that the algorithm considered.

A move is defined as a change applied to one solution to produce another solution. Mobilities are user-defined priorities that are assigned to certain moves. The three possible mobilities are high, medium, and low. A move with a low mobility cannot be implemented by the search algorithm. A move with high mobility can be implemented by the search algorithm. A medium mobility move can be implemented only if it is required to implement a high mobility move. Allowing the user to restrict moves considered by the internal search algorithm reduces the size of the problem. Mobilities prevent parts of the solution from changing, allowing the search algorithm to focus on specific areas of the problem. Mobilities allow humans to use their ability of visual perception and strategic thinking to focus the search algorithm.

The internal algorithm used to find improved solutions could also be selected by the advanced user. There are three choices of search algorithms: greedy, steepest descent, and tabu

(Klau et al. 2002). Allowing the user to select an algorithm during the search process allows for control of the numerical search strategy.

HuGS has been applied to four different applications and various experiments have tested its ability to steer the solution and generate improved solutions. These applications include the traveling salesman problem for package delivery, protein folding, and jobshop scheduling (Klau et al. 2002).

Experimental results from these systems show that humans can effectively steer to a better solution than those generated by an unguided search algorithm. One study concluded that tabu search outperformed other search methods. More importantly, 10 minutes of human-guided tabu search outperformed on average more than 60 minutes of unguided search. Allowing the user to implement combinations of moves, that the computer cannot consider all at once, allows the user to exit local minimums. The user would usually degrade solution quality, but after a few attempts would improve the solution more than if still in the local minimum.

Another MERL experiment used HuGS to compare artificial intelligence heuristics to human-guided searches and to each other. The goal was to solve with the HuGS a Capacitated Vehicle Routing problem with Time Windows (CVRTW). Simple heuristics were created to model some basic human strategies involving reassigning customers to different routes and changing customer mobilities. The human-guided search outperformed all of the heuristics. The best performing heuristic was a greedy random search that could alter the solution to make it infeasible. It would then greedily re-solve and eventually come back to the feasible solution space. The second best performing heuristic randomly selected one of the ten other heuristics during each iteration of the search algorithm. This suggests that using a variety of heuristics might improve solution quality. Overall, the human-guided algorithm was superior to any heuristics used to model human strategies.

2.2.3 UAV Mission Planning

The Draper Laboratory explored the two other solution steering methods: direct and indirect control on the search problem. In addition, Malasky (2005) tested adding HMCDM into a complex optimization system.

2.2.3.1 Mixed-Initiative Control of Automa-teams (MICA)

2LT Jeremy Malasky (2005) applied HMCDM to a complex UAV mission planning program to determine if human-computer collaboration produced superior solutions. In the Mixed-Initiative Control of Automa-teams (MICA), a composite variable formulation was used to select, sequence, and schedule sensing and strike activities of UAVs. The overall problem was broken into five sub-problems: target clustering, selection of aircraft teams to prosecute target clusters, target sequencing, individual route planning, and optimal option selection. MICA used heuristics to solve each of these problems and find a good solution to any given scenario. Malasky incorporated HMCDM in the target clustering and optimal option selection sub-problems. He then compared the HMCDM solution to human only and computer only solutions.

Malasky contended that it would be more efficient to involve humans in the clustering process due to their abilities at spatial reasoning, strategic thinking, and visual perception. Involving the humans in the other sub-problems would allow them to account for situations not considered by the algorithm, and draw on previous experience, tactical doctrine, and intuition. In addition, when involved in the decision process, the humans are more likely to understand and trust the solution. This was especially important in the last sub-problem, where the user selects the optimal options for the final UAV plan. The user can determine which metrics are the most important, perform a risk-reward tradeoff analysis, and understand how the plan will fit into the 'big picture'.

In most cases the value achieved by the HMCDM method was significantly better than or equal in quality to either the human or the computer plans; however, HMCDM plans took on average an additional 3 minutes to generate. The participants commented that they had a better understanding of the HMCDM solutions than they did the other two methods. The users were also more willing to select plans from target clusters that they had generated themselves, and not from computer generated clusters. The experiment concluded that the human provided significant value to the generation of high quality solutions. The visual display of information was important to the users, and in some cases might have reduced solution quality due to poor design. It was determined that the best plans were developed from high quality options and not a large number of options. It is therefore important for decision makers to be presented with high quality options rather than a large quantity of options. The participants attempted different

strategies and had varying levels of success, depending on the scenario. It may therefore be beneficial for teams of operators to be involved in generating solutions.

2.2.3.2 HMCDM UAV Routing

The HMCDM UAV routing experiment (Forest, et al 2007) focused on the direct and indirect control of the search problem. The purpose of the experiment was to determine what level of human-computer collaboration (LOC) was most usable and produced the best results. Users were tasked with routing four UAVs to strike enemy targets. Software developed specifically for this experiment incorporated HMCDM into two of the sub-problems not tested in the MICA experiment: target sequencing and individual route planning. The search algorithm initially clustered the targets by distance, and then implemented a local neighborhood search to find a locally optimal plan. The targets were given attributes of value, risk, and number of munitions required for destruction. The metrics evaluated by the objective function included total target value, risk, percentage of available missiles used (utilization), total distance, and time. An ideal plan would maximize value and utilization and minimize risk, distance, and time. The user was given control over the coefficient values in the objective function. Searching the solution space was possible by selecting metrics weights on a sliding scale from 0 to 100 and implementing the search algorithm. The user could decide on the weights that best matched the given scenario, or use them to escape local minimums in search of improved solutions. The user could also modify plans manually.

The experiment tested four different LOCs. LOC 1 gave the users specific coefficient values and had them implement the search algorithm. The user could then make manual modifications to the plan. LOC 2 consisted of the users determining their own weights and then searching once. The user could then make manual modifications. LOC 3 allowed the user to continually determine and readjust the weights and search, while also having the ability to make manual modifications. LOC 4 provided the user with four plans. The user then selected one plan and could make manual modifications. The four plans were local extreme points for maximum value, minimum time, minimum risk, and minimum utilization.

Experiment participants found LOC 3 to be the most supportive of their ability to find an optimal plan. LOC 2 and 4 were tied for the most usable, and took the least amount of time to

plan. The users found it easier and more intuitive to adjust the weights than manually modify the plans. In an interview after the experiment, a participant commented that he trusted the solution to LOC 3 the most because he was confident he could not improve the solution after numerous searches. He also found LOC 4 helpful because it gave him a basis of comparison for the extreme points of the local search. As the users made manual adjustments to the plans, the computer would calculate the updated plan metrics. Some users found manual modification too difficult because it was not easy to improve the solution and predict how the metrics would change. This caused them to rely on the computer's search algorithm. A participant suggested that providing four plans to select from and then allowing multiple searches with adjustable coefficients would be a desirable option. This is similar to the capability of the HuGS software, which allows the user to select, modify, and steer a selection of pre-computed plans.

The scenarios provided to the participants encouraged them to maximum plan value while minimizing plan time. Discussions with the participants revealed that they also included expected UAV attrition, utilization, and distance into their choice of an optimal plan. Users also determined risk tolerances that influenced their plan selection. In comparing the plans selected in the four LOCs by time and value, the best plans were created by LOC 1, followed by LOC 2, LOC 3, and LOC 4. It is likely that plans worse in time and value were selected due to better attrition or distance metrics. This highlights the difficulty of comparing plans when humans have different internal objective functions. Another important observation was that some users had trouble understanding the local nature of the search and the need to search the solution space to escape local minimums. Overall, the experiment demonstrated the utility of using adjustable coefficient objective function weights to search the solution space for complex UAV routing scenarios.

2.2.3.3 Decision Space Visualization

Decision Space Visualization (DSV) software was developed to augment the HMCDM UAV routing software (Forest, et al 2007). The goal of the DSV software was to allow the user to view the entire decision space and easily compare plan metrics. Plans metrics were graphed on a two dimensional plot. The user could select the metrics for each axis and for a color scale, allowing three metrics to be visually displayed. Plans on the local Pareto frontier for the two axes were highlighted. The user had the ability to filter plans by the percentiles of the metric

values. When a plan was selected on the plot, all plan metrics were displayed on a side panel of the interface. It was possible for the user to display a visual representation of the plan. The user could also highlight and take notes on specific plans for future reference while searching the solution space. The highlighted plans could be compared side by side with graphical and numerical data.

A cognitive walkthrough was conducted to test the software's ability to complement human decision making in a UAV routing experiment. In the cognitive walkthrough, the participants were given a scenario identical to one of those used in the HMCDM UAV routing experiment. The plans plotted with the DSV were those created by participants during the HMCDM UAV routing experiment. The observer noted how the user implemented the software to pick an optimal plan. The observer also asked questions during the experiment to probe the participants' thought process.

The two users, both with operational experience in the U.S. Army, used the software very differently. Participant 1 (P1) picked a plan by evaluating the metrics and comparing tradeoffs between time, value, expected value, and attrition. He made use of the ability to set the axis and color scale and the highlighted Pareto frontier. Using the plan comparison he narrowed down his selection to a final plan based on the metrics. Participant 2 (P2) was interested in developing a plan that would destroy targets by clearing a corridor for Army units to penetrate the enemy city. His plan was evaluated based on visually comparing the plans to determine which plans were closest to the plan he had envisioned. He methodically looked at all of the plans by displaying them and noting which ones matched his envisioned plan. He then selected his axis and used the Pareto frontier plans as a comparison to the ones he previously selected. He finally selected a plan that was far from the Pareto frontier, but best cleared a corridor for invading troops. While P1 relied on the plan metrics to pick a plan near the Pareto frontier for time and distance, P2 relied on the plan display and then compared it against plans along the Pareto frontier. Both were confident that they made the best use of their UAVs for the given scenario.

P2 mentioned that being able to use the HMCDM UAV routing software to select plans with his desired corridor and use the DSV software to compare the metrics could be useful. This would require the HMCDM routing software to generate multiple plans similar to what P2 envisioned and then use the DSV software to compare them.

P1, as a previous participant in the HMCDM UAV routing experiment, had much more confidence in the plan that he selected with DSV software than the one he generated in the previous experiment. In the HMCDM experiment, he felt that he was spending too much time building his own plan that was likely not to be optimal. He trusted that the plans in the DSV represented all of the possible plans in which he would be interested. Depending on how plans are selected to be in the DSV software, this assumption might not be true. P1 mentioned that the HMCDM UAV routing software would be good for high-level commanders to generate possible solutions to the problem. Those solutions could then be passed to operators in the field using the DSV software to analyze closely the metrics and select the best plan. The HMCDM UAV routing software could then be used for slight modifications of the plan.

When questioned about the metrics' accuracy, P2 said that he trusted that the metrics were calculated correctly. P1 said that he would like to know the accuracy of the metrics, and that it would add another layer to his decision-making process, but not ultimately change how he would find the best solution. With inaccuracy of data, and the fact that mathematical models may have errors, providing metrics accuracy and calibrating trust in the software may prevent the user from assuming the metrics are exact and may even affect the users' decision-making process.

Overall, the experiment showed the utility of both the HMCDM routing and DSV software. Depending on both the scenario and user, the two software packages may be useful to planning a UAV mission. Linking the two packages into one may improve the ability for the user to generate quality plan options and compare them. However, this capability may not be for just one decision maker to generate plans and pick the best one; the chain of command could use the software iteratively to pass options down to field operators who would select the best plan. The MICA, HMCDM UAV routing, and DSV experiments highlight the promise of HMCDM software to benefit decision making.

2.2.4 HGA Considerations

The benefits of involving the human in the search process include incorporating dynamic human objectives, heuristic methods, and knowledge that are not easily captured in an optimization model, altering the problem definition to match the changing constraints of the real world, and

focusing computational power on promising areas of the solution space. HGAs are likely to be beneficial in time-critical or complex scenarios where the user needs to quickly generate high quality solutions to a changing problem or cannot process the full range of possible solutions. An HGA is designed to be understood and used by decision makers, not experts in Operations Research. To accomplish these goals the HGA must be built around both the problem and the human.

The solution quality of the HGA depends on many factors that might reduce/enhance the quality of the solution. These factors include user experience with the HGA, knowledge of the problem domain, and time spent searching the solution space. Non-HGA solution quality is usually known to be optimal, relative to the defined problem model, or solved with heuristics to within a known percent of optimal solution quality. This makes the HGA solution quality more variable than the quality of non-HGA solutions. However, a non-HGA is sensitive to the defined problem model and cannot easily be extended to a range of problems. If the non-HGA solution does not incorporate all of the human objectives and knowledge of the problem and solution space, the non-HGA solution may not be the best solution to the real world problem. The HGA has the ability to consider the real world implications of the solution through the human operator, even if the internal algorithms do not capture this knowledge. HGAs are designed to be more flexible than non-HGAs. Ideally, the HGA should be able to hone in on the best real-world solution, but the factors affecting the variability of solution quality must be investigated for HGAs to become a feasible option.

The HGA is controlled by a human operator or team of operators. A computer uses the human inputs to follow a scripted set of commands (an optimization or heuristic algorithm) that search the solution space to find the solution to the given operator defined problem. Through the four methods of solution steering, the user can alter/repeat this process until he/she is content with the final solution. The human relies on these underlying algorithms in the HGA search, but has more control over the problem definition, when/how the non-HGAs operate on the problem, and what direction the non-HGAs search. Since the HGA is dependent on the non-HGAs used in the computerized search process, the HGA solution quality depends on the quality of the internal non-HGAs. Adding human input into non-HGAs should allow the user to capitalize on the strengths of those algorithms while compensating for their weaknesses. An HGA should be used

when human input into the solution process can augment the power of the internal non-HGAs. When this is not the case, the non-HGAs alone will be the better method for solving the problem.

A user that is actively involved in the HGA solution process is more likely to understand and appropriately trust the final solution than non-HGA solutions. Decision makers are more likely to be skeptical of computer generated solutions when they do not understand how/why the solution was generated. Allowing the decision maker to become involved in the search process improves the user's understanding of the solution, therefore fostering appropriate trust in the HGA and the solution.

2.3 Trust

J. Roseborough describes his dilemma: "In any system requiring a human operator, the objective validity of a specific decision aid can never be established if the decision aid is intended to be used for decisions requiring information that is not explicitly model-able" (qtd. in Sheridan 2002). The logic of the Roseborough Dilemma is described by Sheridan (2002) in five steps:

1. In a complex control system, the controlled process cannot be fully and explicitly modeled, nor can the objective function.
2. Therefore, one appropriately falls back on the human operator to compliment whatever mechanized embodiment of these functions is provided.
3. Given that the human operator makes mistakes, it is evident that he or she can be helped by a computer-based decision aid, so one tries to provide such an aid.
4. Ideally, to design the decision aid and evaluate the human operator's use of it, a relatively complete process model and objective function must be used as a norm. However, Step 4 is in conflict with Step 1.
5. Further, if such a relatively complete process model and objective function were available, then why not use these in place of the human operator to provide an automatic decision maker, thus leaving out the human?

Sheridan (2002) describes three assumptions required to avoid the dilemma:

1. The human decision maker is necessary for the information that is not explicitly modelable. No valid decision aid can be built to provide such information when it is needed.
2. In some, perhaps most, decision situations the human operator will encounter, he or she will require only information that is modelable. The human will always make some mistakes in such decisions and can benefit from a decisions aid for these cases, and in such cases the decisions aid can be validated. (So it makes sense to provide such a decision aid for those situations.)
3. *The human can properly decide when the situation included elements the decision aid can properly assess and can know for which elements the decision aid should be ignored.*

Sheridan (2002) points out that the third assumption "is a big one, and it is an issue in obvious need of research." There are always conditions in which highly reliable automation will fail. Failure can be caused by the noisiness of the real world, unplanned variations in operating conditions, unexpected or erratic behaviors of other system components or human operators, system malfunctions, and the inherent unreliability in predicting the future (Parasuraman, Sheridan, & Wickens 2000). The key to the third assumption is ensuring the user appropriately trusts or distrusts the decision aid, depending on the situation.

2.3.1 Definitions

Researchers have defined trust with respect to automation and determined its components in various ways:

- "A generalized expectation related to the subjective probability an individual assigns on the occurrence of some set of future events" (Rempel qtd. in Muir 1994).
- "The degree of confidence you feel when you think about a relationship" (Rempel & Holmes qtd. in Muir 1994).
- "The confidence that one will find what is desired from another, rather than what is feared" (Deusch qtd. in Muir 1994).
- "A generalized expectancy held by an individual that the word, promise, or written statement of another individual or group can be relied on" (Rotter qtd. in Muir 1994).

- "The attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability" (Lee & See 2004).

- "The expectation held by a number of a system, of the persistence of the natural and moral social orders, and of technically competent performance, and the fiduciary responsibility, from a member of the system, and is related to but not necessarily isomorphic with, objective measure of these properties" (Barber qtd. in Muir 1994).

- Sheridan (2002): cause and effect, judged reliability of a system, perceived robustness, demonstrated or promised ability, familiarity, understandability, and usefulness.

- Zuboff (Muir 1994) defines three aspects of trust: trial and error experience, understanding of technology, and faith.

2.3.2 Models and Experiments

The present section provides an overview of trust theory as it relates to automation. Starting with Barber's three dimensions of trust and building to the Lee and See model, various research has contributed to current trust theory.

2.3.2.1 The Three Dimensions

Muir (1994) summarizes Barber's development of trust theory. Barber breaks trust into three dimensions: persistence, technically competent performance (competence), and fiduciary obligations and responsibility (responsibility). The first dimension, persistence, includes three components that explain how the world works: natural physical order, biological order, and moral social order. These orders form the foundation of trust, limit possible outcomes by the way that things work, and allow for the creation of mental models and prediction of future states. The second dimension, competence, includes three types: expert knowledge, technical facility, and everyday routine performance. This dimension is the most relevant to a human having trust in a machine. The third dimension, responsibility, is the duty in certain situations to place others' interest above your own. This forms the basis of trust when the user's technical competence is exceeded by the automation. This includes the operator's expectation that the automation will meet design criteria when the automation has superior knowledge, autonomy, authority, power, or unknown competence. Barber defines responsibility as a property of the referent (the automation) when assessment of its competence cannot be made.

2.3.2.2 Dynamic Model

Rempel et al, as explained by Muir (1994), create a dynamic model based on predictability, dependability, and faith. The development of trust starts with predictability, progresses to dependability, and finally reaches faith. Rempel et al. believe that trust builds in this order and breaks down the same way. Predictability depends on consistency and desirability of three factors: actual predictability of machine behavior, operator's ability to estimate machine predictability, and stability of the environment in which the system operates. Machine transparency and user experience improve the operators estimation of machine predictability. Since a reliable system may not be reliable in an unstable environment, it is necessary for the user to recognize unstable environments. If the user can appropriately trust the machine in a given environment, the machine is less likely to be distrusted and disused in a stable environment.

Dependability is the extent to which a machine can be relied on. This is especially important in risky situations or unstable environments. Dependability is based on the accumulation of evidence that supports predictability. Increased knowledge of the machine and pushing the machine beyond its limits might improve the user's dependability assessment.

Faith, the ultimate level of trust, is based on predictability and dependability. It is the closure against doubt, based on an uncertain future. It is an expectation of the user supported by extended and varied experience. Faith depends on the user's perception of the software's flexibility and appropriateness.

2.3.2.3 Muir Model

Bonnie Muir (1987, 1994, Muir & Moray 1996) explored trust theory and conducted various experiments to test her ideas. According to Muir, trust theory should be able to explain the nature of human trust in machines, how trust changes with increased user experience, and the relationship between user trust and automation use. She also has identified various attributes of trust. One attribute of trust is that it is an expectation of another, oriented towards the future, with the purpose of prediction. Trust has a specific referent, and those referents can be trusted to various degrees. Trust can also relate to different properties of the same referent.

Muir's definition of trust is mathematical. She linearly formulates Equation 1 to define user trust by summing the expected values of Barber's three dimensions.

$$T_i = E_i[P_n + P_m] + E_i[TCP_j] + E_i[FR_j]$$

i = subjective values determined by the user, might vary from true properties
j = referent (the automation)
Pn= probability of natural order
Pm = probability of moral social order
TCP = technically competent performance
FR = fiduciary responsibility

Equation 1: Muir Linear Trust Formulation
(Muir 1994)

A more complex formulation, Equation 2, can account for the multiplicative effects of the variables and the need for weights to adjust their relative importance to the overall trust calculation.

$$T_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_2 + \beta_5 X_1 X_3 + \beta_6 X_2 X_3 + + \beta_7 X_1 X_2 X_3$$

$\beta_{0\text{-}7}$ = coefficients
X_1 = E[persistence]
X_2 = E[technically competent performance]
X_3 = E[fiduciary responsibility]

Equation 2: Muir Trust Formulation with Multiplicative Effects
(Muir 1994)

Muir breaks down trust into different components by placing Rempel et. al. and Barber's dimensions of trust on orthogonal dimensions (see Table 2). The matrix is intended to apply to many applications but be detailed enough to characterize a specific situation. The expectations of persistence, competence, and responsibility are still used to define trust, but they are determined by the nature of the relationship between the user and automation. In the early stages of the relationship, persistence forms the basis of trust. Trust then progresses to dependability and finally to faith in a mature relationship. At least one cell of the matrix should describe a system at any point in time.

Basis of expectation at different levels of experience			
Expectation	**Predictability (of acts)**	**Dependability (of dispositions)**	**Faith (in motives)**
Persistence			
- Natural physical	Events conform to natural laws	Nature is lawful	Natural laws are constant
- Natural biological	Human life has survived	Human survival is lawful	Human life will survive
- Moral social	Humans and computers act "decently"	Humans and computers are "good" and "decent" by nature	Humans and computers will continue to be "good" and "decent" in the future
Technical competence	j's behavior is predictable	j has a dependable nature	j will continue to be dependable in the future
Fiduciary responsibility	j's behavior is consistently responsible	j has a responsible nature	j will continue to be responsible in the future

Table 2: Muir Framework for Studying Trust
(Muir 1994)

Muir models in Figure 3 the relationship between automation and user trust. In the model, referent *j* (*j* can be the automation as a whole or a specific component) has an actual competence, responsibility, and other attributes. These attributes formulate the true behavior of *j*. The behavior feeds into operator *i*'s mental model of referent *j*. Operator *i* has various mental models, including one of himself/herself, that have influence over each other. Each mental model contains the user's perception of persistence, competence, and responsibility (Barber's dimensions). Each perception is based on predictability, dependability, and faith (Rempel's dimensions). In other words, the perception of *j* comes from the applicable cells in the Barber/Rempel Matrix. These perceptions lead to expectations of persistence, competence, and responsibility which all feed into user *i*'s trust in *j* (Muir's linear model). At this point, user trust can be compared to the actual attributes of *j* to determine how well the trust is calibrated. Trust then leads to a prediction of *j*'s behavior and a confidence in the prediction. The confidence may be influenced by the plausibility of the prediction and the user's mental model of his own abilities. After *j*'s behavior occurs and *j* provides the user with feedback, the accuracy of the prediction is evaluated and trust is calibrated. It is important to note that appropriate feedback of *j*'s behavior must be provided to *i* in order to calibrate trust correctly.

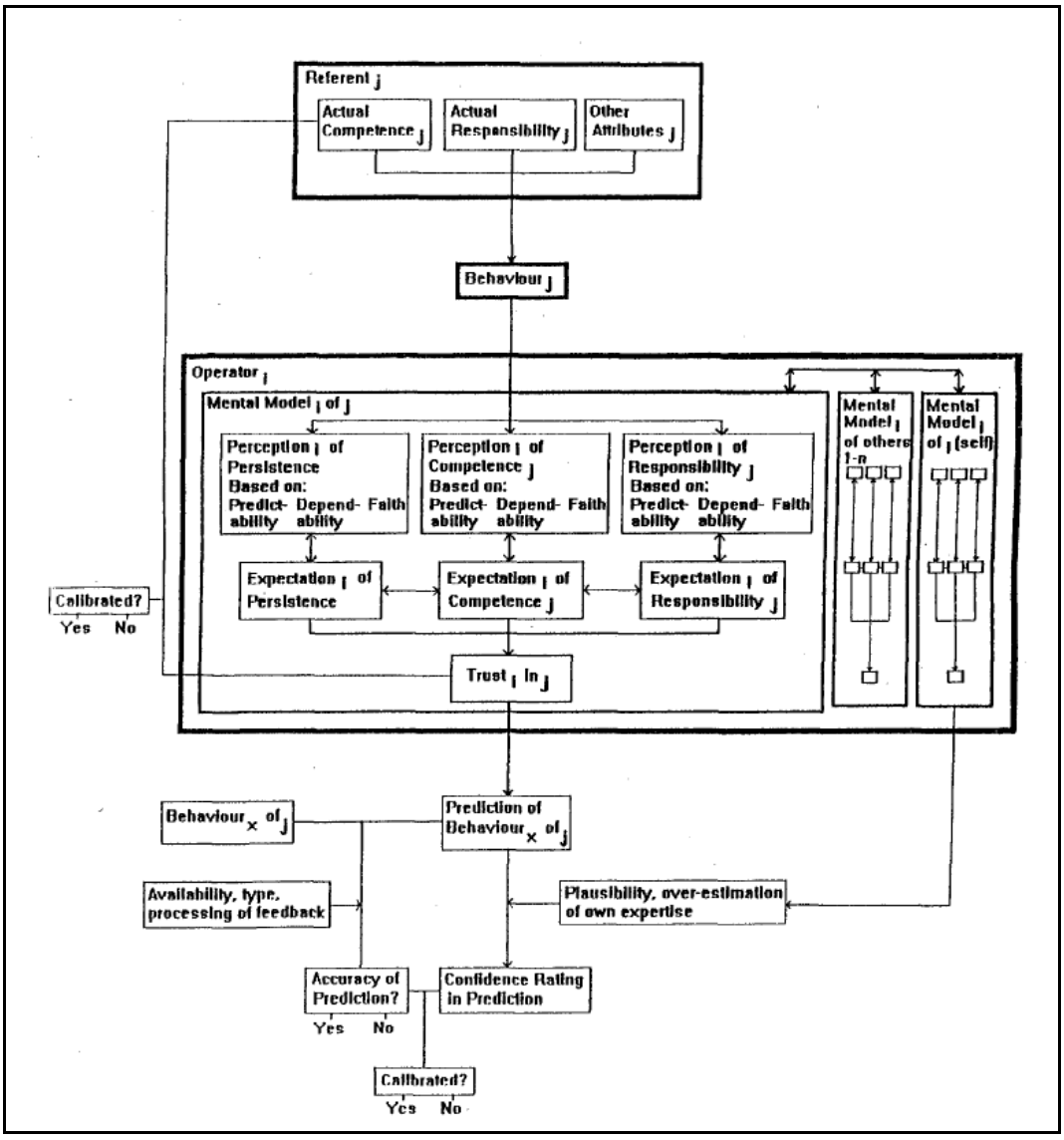


Figure 3: Muir Model of Trust Calibration
(Muir 1994)

Muir defines distrust as not the absence of information, but the expectation of incompetence or irresponsibility. Distrust leads to disuse of automation, which does not give the operator an opportunity to regain trust in the system.

Muir tested some of her ideas in two experiments (Muir 1996). The experiments explored user trust as they operated pumps in a simulated milk pasteurization plant. The first experiment evaluated competence and responsibility in relation to trust and discovered that competence was the best predictor of system trust. Competence was defined as how well a system component performed, given its design. In addition, it appeared in early stages of the

experiment, that faith was the most influential factor. Faith was then followed by dependability and predictability, which was the opposite development proposed by Rempel. The experiment also compared errors made by pumps simulated in the automation. Another result showed that small variable errors were just as detrimental to trust as large constant errors. Distrust spread to independently functioning components of the system, but it did not spread to the entire system.

In the second experiment, Muir concluded that the magnitude of the pump error affected trust. Small errors significantly reduced trust. Larger errors reduced trust more than small errors, but at a decreasing rate as the errors grew larger. Users initially had conservative trust in the pumps: trust in accurate pumps increased over time and trust in inaccurate pumps decreased. Muir concluded that users base trust on specific, immediate, negative consequences and not long term positive consequences. Therefore, machine behavior must be consistent and desirable for the user to trust it, even if negative short-term machine behavior does not affect the long-term result. Muir concluded that trust is fragile: a user's temporary loss of automation control can diminish trust, without destroying it. This is called brittleness.

2.3.2.4 Extension of the Muir Model by Lee and Moray

John D. Lee and Neville Moray (1992, 1994) extend Muir's theory. Instead of placing Barber and Rempel's trust dimensions on orthogonal axis, Lee and Moray assume they are complementary. They define four dimensions of trust that combine the theories of Barber, Rempel, and Zuboff: purpose, process, performance, and foundation (see Table 3).

	Barber	**Rempel, Holmes, and Zanna**	**Zuboff**
Purpose	Fiduciary responsibility	Faith	Leap of faith
Process		Dependability	Understanding
Performance	Technically competent performance	Predictability	Trial-and-error experience
Foundation	Persistence of natural laws		

Table 3: Lee and Moray Trust Dimensions

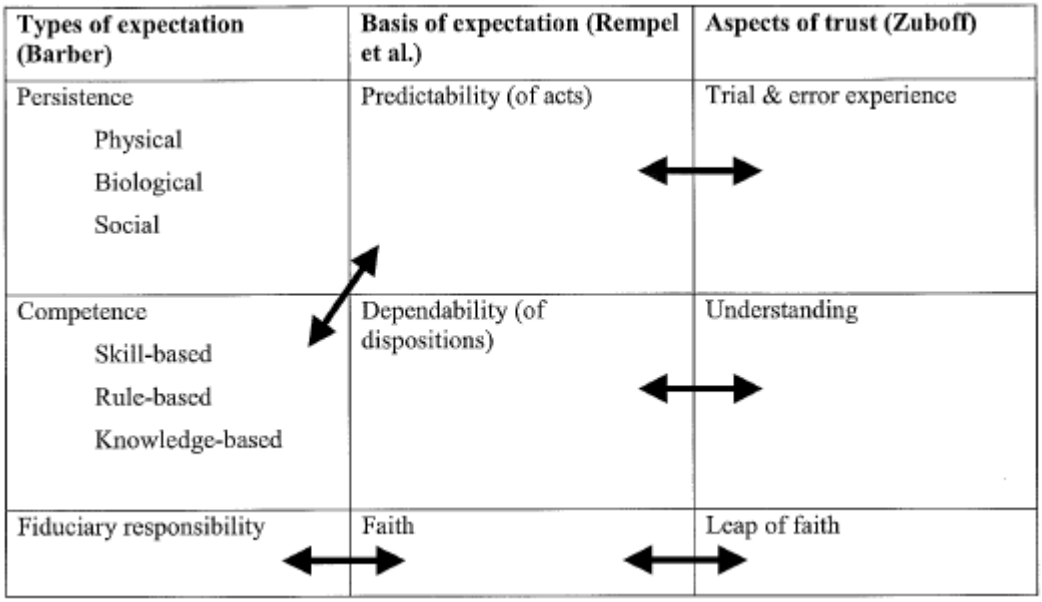


Figure 4: Condensation of Various Model to Form Lee and Moray Trust Model
(Cohen, Parasuraman, & Freeman 1998)

Purpose includes fiduciary responsibility, faith, and leap of faith. Process includes dependability and understanding. Performance depends on technically competent performance, predictability, and trial and error experience. The foundation of trust depends on the persistence of natural laws. Figure 4 shows how Lee and Moray condensed the Barber, Rempel, and Zuboff trust models.

Lee and Moray's experiment (1992) attempted to improve upon Muir's experiments with a simulated orange juice pasteurization plant experiment. In the experiment, trust increased as the users gained experience with the system. As the system made errors, user trust decreased, but recovered over time. Both trust and performance had learning curves. The experiment was designed so that faults made by the system did not affect the long-term performance. When the system made chronic faults, performance recovered before user trust. Lee and Moray used their data to develop a causal and dynamic trust model. The causal model using linear regression was used to describe the factors that influence trust. It was found that the occurrence of fault and system performance were the only two significant variables to forming trust. The model accounted for 53.5% of the variation in the level of trust and had highly correlated residuals. A dynamic model, Equation 3, was created using an autoregressive moving average vector form (ARMAV) to show how trust changes over time. The ARMAV model accounted for 79.1% of the variation in trust and did not have correlated residuals.

$$\text{Trust}(t) = \beta_1 \text{Trust}(t-1) + \alpha_1 \text{Performance}(t) + \alpha_1 \beta_2 \text{Performance}(t-1) + \alpha_2 \text{Fault}(t) + \alpha_2 \beta_3 \text{Fault}(t-1)$$

Equation 3: Lee and Moray ARMAV Trust Model 1
(Lee and Moray 1992)

Most users in the experiment attempted a feed-forward control strategy. In a feed-forward strategy, the user inputs control at the beginning of the experiment based on their mental model of the automation, and lets the automation run smoothly until the end of the trial. The users preferred feed-forward manual control, but when a system fault occurred, they switched control to automatic mode in an effort to focus their efforts on fixing the error. A fault in manual mode reduced the users' self-confidence, causing them to allocate automatic control. System faults actually increased the allocation of automatic control. After the fault, a feedback strategy was adopted. With a feedback strategy, the users operate the system in response to its actions in order to fix the errors.

In a second experiment, Lee and Moray compared the affects of self-confidence and trust on automation allocation between manual and automatic control. The experiment revealed that the users generally had overconfidence in their abilities, but users with inappropriately low self-confidence were more likely to rely on automation. User trust in automation had to be greater than their self-confidence for them to allocate automatic control. This minimum difference between trust and self-confidence required for automatic control was referred to as bias. The bias may have been due to the users' exploratory behavior to test their abilities in manual mode during the beginning of the experiment. Users were more likely to rely on automation that was more transparent. An ARMAV model, Equation 4, was created to explain the percent of time the users allocated automatic control:

$$\%\text{Automatic}(t) = \beta_1 \%\text{Automatic}(t-1) + \alpha_1 ((\text{T-SC})(t)) + \alpha_1 \text{Individual Bias} + a(t)$$

%Automatic: Percent time in automatic control
T: Trust
SC: Self-confidence
a: *normally distributed error*
β, α: *coefficients*
t: time

Equation 4: Lee and Moray ARMAV Trust Model 2
(Lee & Moray 1994)

The ARMAV model accounted for 60.9%, 80.5%, and 86.5% of the variation of trust in three pumps. The model revealed that in addition to trust and self-confidence, operator bias and previous automation use had a significant impact on current automation use.

2.3.2.5 Errors, Automation, & Trust

Skitka, Mosier, and Burdick (1999) tested user trust in an automated device by measuring the number of errors made by users relying on the automation. Commission errors are mistakes made when a user follows the directives of automation when other information shows that the automation is incorrect. Omission errors are mistakes made when a user does not perform a necessary action because the automation did not say to do so, despite non-automated indications that the action should be performed. The experiment showed that users were prone to making both of these errors due to overtrust in the automation. Users made more omission errors when using automation than when not using automation. Most users felt that the automation did not reduce their workload. Participants who thought the automation was making few errors were more likely to be making errors themselves and delegate responsibility to the faulty automation. Users with perfectly working automation made fewer errors than with faulty automation. The use of automation resulted in less vigilant monitoring and users taking the cognitive path of least resistance by following automation directives without checking other reliable evidence to help in their decision making.

2.3.2.6 Reliance on Automation

Dzindolet, et. al. (2003) conducted three studies to explore the role of trust in automation reliance. The automation was a computer that scanned an image to determine if a camouflaged soldier was hidden in terrain. The user also attempted the same task and compared performance. The automation was programmed to make half as many errors as the user. In the first experiment, participants had a positive bias towards the automation's capability, but this trust bias was statistically insignificant from their own level of self-confidence. Initially, most users assumed that the automation performed better than they did. The users could view the automation's conclusion after every image and therefore observe if it made mistakes. When users observed the automation make simple errors during the first experiment, they began to

disuse the automation even though it was superior to their own ability. It was concluded that since the automation's mistakes were inconsistent with their initial expectations, the user remembered the mistakes, which led to disuse.

In the second experiment, the users could only see the automation's cumulative success after every five images. This prevented users from identifying when the automation made 'easy' errors. This method of preventing the user from seeing the automation's obvious errors eliminated the disuse.

The final experiment provided the user with an explanation of why the automation made certain mistakes. Users were provided with automation that was either superior or inferior to them. In both cases, the users were more likely to trust the automation when provided with the extra information. Users trusted the superior aids more than the inferior aids, but both were relied on equally. For the superior automation, this resulted in appropriate reliance, but for the inferior automation, this resulted in misuse. It was also concluded that, "comprehensive instruction on the algorithms used by the aid may be effective in explaining not only why the aid might err, but also how the aid arrives at correct decisions."

2.3.2.7 Malfunction and Automation Allocation

Itoh, Abe, and Tanaka (1999) explored user allocation of automation after various malfunctions. In automatic mode, five continuous malfunctions decreased trust more than five time-independent mistakes. Inexperienced users were slower than experienced users in switching from manual to automatic control after the automation stopped malfunctioning. It was concluded that inexperienced operators were confused on how to interact with the automation during malfunctions. It took longer for inexperienced user trust to recover, resulting in disuse.

2.3.2.8 Lee and See Model

John D. Lee and Katrina See (2004) create a dynamic trust model. Figure 5 shows the relationship between trust, situational context, reliance, and automation. They build upon previous research and provide a well researched and one of the most complete and current theories on trust. They define trust as an attitude (see page 45) and not a belief, intention, or behavior. A belief influences trust, which then creates an intention and finally a behavior. Trust plays a crucial role in this closed loop process, explaining human interaction with automation.

Four components of the model must be explained: appropriateness of trust, situational context, basis of trust, and the cognitive process of trust.

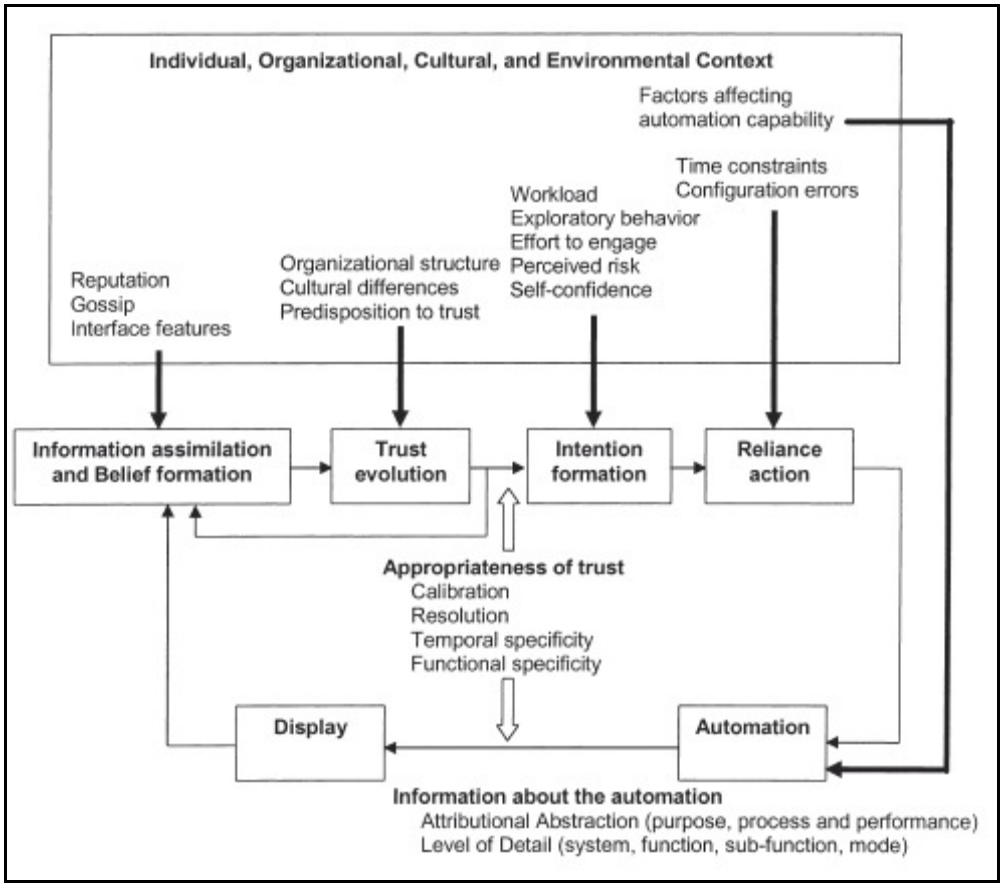


Figure 5: Lee & See Dynamic Trust Model
(Lee & See 2004)

The appropriateness of the trust depends on four factors: calibration, resolution, and temporal and functional specificity. Appropriateness is defined as the "relationship between the true capabilities of the agent and the level of trust" (Lee & See 2004). Calibration is the "correspondence between a person's trust in the automation and the automation's capabilities" (Lee & See 2004). Resolution measures how precise a judgment of trust differentiates levels of automation capability. For example, if a large change in automation ability corresponded to a small change in user trust, the precision would be poor. Specificity is the degree to which trust is associated with a particular component. Functional specificity refers to the individual components of the system. A user with low functional specificity trusts the entire system as a whole. Someone with high functional specificity would trust individual components of the

system to various degrees. Temporal specificity refers to the sensitivity of trust to changes in context that affect automation capability. A user with high temporal specificity trusts the system differently at appropriate times, depending on the situation. A user with good calibration, high specificity, and high resolution can match the capabilities of a system component at a specific time to an appropriate level of trust and therefore help prevent misuse or disuse. Figure 6 shows the relationship between trust and automation capability.

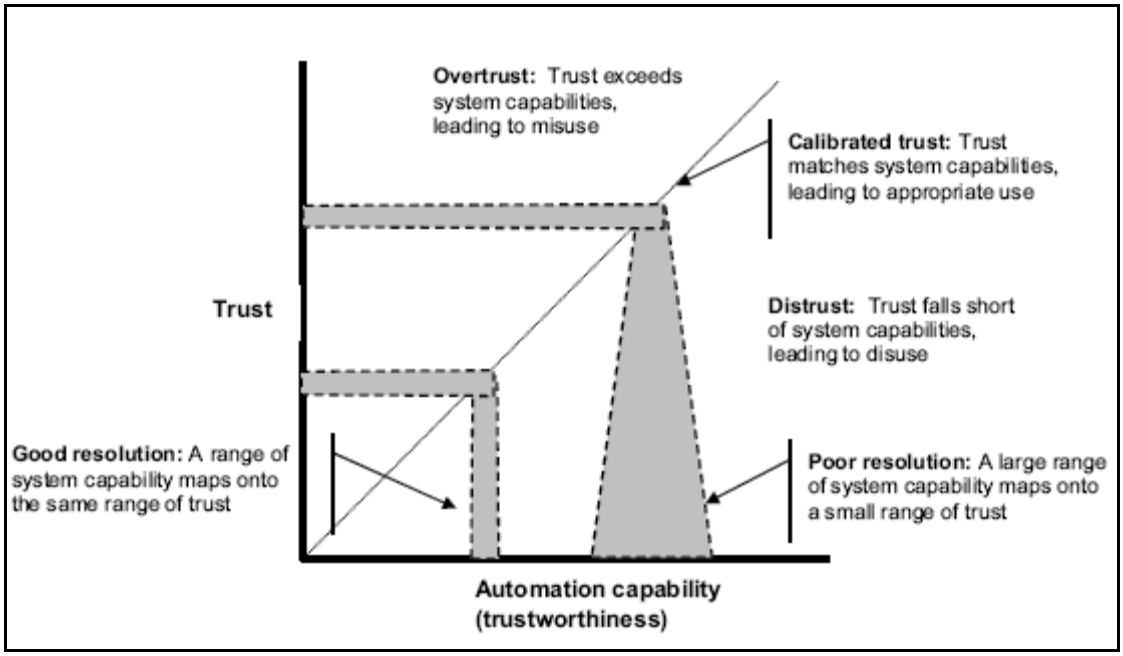


Figure 6: Lee and See Graph of Automation Capability and Trust
(Lee & See 2004)

The context of trust depends on individual, organizational, and cultural contexts. The individual context is determined by the user's past experience and natural tendency to trust. Users can trust differently, even when faced with identical scenarios. The organizational context depends on gossip, reputation, and formal and informal roles that a user or automation play in an organization. Cultural context is influenced by social norms and long-term experience with certain groups of people. For example, airline pilots might trust differently than non-pilots.

The basis of trust is the "information that informs the person about the ability of the trustee to achieve the goals of the trustor" (Lee & See 2004). This basis is based on attributional abstraction and level of detail. Attributional abstraction refers to the dimension of trust. Lee and See (2004) refined Lee's earlier work with Moray (1992, 1994) by defining only three

dimensions: performance, process, and purpose. Performance is the current and historical operation of the automation, which includes reliability, predictability, and ability. Process is the degree to which the automation's algorithms are appropriate for the situation and can meet the operator's goals. The purpose is the reason the automation was developed and how it is used with respect to the designer's intent. Trust depends on the observations and inferences made between the dimensions. These dimensions are not stages of development, but attributes of trust. Level of detail refers to the requirement that specific information is needed to promote high functional specificity of the overall system and its components. Availability of information at various levels of detail and attribution abstraction can foster appropriate trust.

The cognitive process explains how people use information to form the basis of trust. The analytic, analogic, and affective processes interact to form trust. The interaction is dependent on the relationship between the trustor and trustee, available information, and how the information is displayed. In the analytic process, the user draws from past experience to form a mental model of the expected costs and benefits of making a decision. This is a cognitively demanding knowledge-based process that requires conscious calculations and comparison of alternatives. The analogical process is rule-based and less cognitively demanding. It matches trust to agent characteristics and the environmental context. Rules and procedures, observations, information from intermediaries, and condition-action pairings form the analogical process. The affective process is based on emotion and has the most influence on trust and corresponding user behavior. When forming a cognitive model is too complex or rules do not exist to govern a situation, emotion is used to guide behavior. Lee and See theorize that in order to create appropriate trust, information must be presented in a manner compatible with the analytic, analogic, and affective processes. This is similar to the theory behind Ecological Interface Design, presented later in this chapter.

Some additional critical elements of the dynamic model include the closed loop dynamics of trust and reliance, the role of the situation and environment on trust and reliance, and the importance of information display on appropriate trust. The closed loop dynamics make trust dependent on automation use. Relying on automation makes it possible for the user to understand it and improve their trust in it. Automation that is not used is less likely to develop improved user trust. Nonlinear relationships between trust and reliance can develop in a closed loop system, which make trust difficult to explain and predict among different users. In addition,

trust and reliance have inertia, meaning that trust is dependent on previous experience and not entirely the current situation. Situational factors such as, situational awareness, self-confidence, workload, perceived risk, and effort to engage or explore the automation affect trust and reliance. Finally, the display is essential to providing information about the three dimensions of trust. To develop appropriate trust the information must be displayed in the correct format and level of detail.

Lee and See (2004) discuss the difference between trustworthy and trustable automation. Trustworthy automation performs efficiently and reliably, often with very complex and hard to understand algorithms. Trustable automation "supports adaptive reliance on the automation through high-calibration, high-resolution, and high-specificity trust" (Lee & See 2004). Trustable and trustworthy automation can be competing objectives in automation design. There are situations when trustworthy automation should be less complex and less capable for the sake of making it more trustable. Trustable automation is more likely to be used correctly due to appropriate trust.

2.3.3 Calibration

Sheridan (2002) describes calibration as "enough trust to make effective use of advice and enough distrust to stay alert and know when to ignore it." Over-trust results when user trust is greater than automation capability. Under-trust results when user trust is less than automation capability. The purpose of calibration is to make trust match automation capability. Poor calibration can have serious consequences. When good automation is overrode or disregarded due to under-trust, type I error results. In this case, the user loses the benefits of automation, experiences increased workload, and possibly makes more errors. Over-trust results in type II error, which is when a user fails to override faulty automation. Type II error can be more dangerous that type I because it can result in automation shutdown and an increased chance of a serious accident.

Muir (1994) recommends various methods to improve calibration with respect to general automation and automated decision aids:

- Elements of a system should be calibrated independently.
- Identify poorly calibrated referents and determine if the problem lies with the expectation of persistence, competence, or responsibility. Retrain on the problem expectations.
- Provide objective data about the automation:
  - Performance over time
  - History of behavior
  - System constraints
  - Environmental disturbances
  - Domain of competence
  - Explicit criteria for acceptable levels of performance
- Improve system transparency.
- Match transparency with user expectation of trust: novices need more transparency to aid in appropriately trusting the system.
- Educate users on system intentions and responsibility.
- Allow operator to experience automation in risky and unfamiliar situations.
- Give user time to explore the system and develop accurate expectations.
- Introduce automation carefully to prevent mistrust and need for recalibration.
- Users should be continuously calibrated and be aware of their attitudes towards trust.

Muir (1994) concludes that the "well calibrated use of an instrumental decision aid maximizes overall system performance in decision making and problem solving by readily accepting output of competent machines and relying on his own competence or other resources for function which the machine handles improperly."

Lee and See (2004) provide ideas on how to make automation more trustable and how to relate situational context to automation capability:

- Making Trustable Automation
  - Design for appropriate trust, not greater trust.
  - Show automation past performance.
  - Show process and algorithms of the automation by revealing intermediate results in a way that is comprehensible to the operators.

- Simplify the algorithms and operations to make it more understandable.
- Show the automation's purpose, design basis, and range of applications in a way that relates to the user's goals.
- Train operators on expected reliability, mechanisms governing its behavior, and intended use.

- Relate Context to Automation Capability
  - Reveal the context and support the assessment and classification of situations relative to the capability of the automation.
  - Show how past performance depends on situations.
  - Consider how environmental context influences the relevance of trust; the final authority for some time-critical decisions should be allocated to automation.
  - Evaluate the appropriateness of trust according to calibration, resolution, and specificity. Specificity and resolution are particularly critical when the performance of the automation is highly context dependent.
  - Training to promote appropriate trust should address how situations interact with the characteristics of the automation to affect its capability.

2.3.4 Trusting Decision Aids

Cohen, Parasuraman, and Freeman (1998) create a trust model for decision aids. They focus on fostering appropriate trust in uncertain domains of the decision aid. Their theory, called the Argument-based Probabilistic Trust (APT) model accounts for the qualitative and quantitative aspects of trust. They define trust as "a measure of uncertainty regarding system performance" (Cohen, Parasuraman, & Freeman 1998). Toulmin's theory of argument is used to explain the qualitative nature of trust. A claim is a conclusion that is being evaluated. The evidence that supports the claim is called grounds. A warrant is the belief that connects the grounds to the claim. The backing provides a theoretical or empirical basis for a warrant. Modal classifiers describe the claim's validity as strong or weak. Rebuttals are conditions that can invalidate the warrant. Figure 7 shows the ATP model based on Toulmin's argument theory.

The APT model applies five interrelated parameters to describe an assessment of trust:

1. Temporal Scope: Duration of time that the claim covers. There are four phases in the use of a decision aid:
   - Trust in a system over all potential uses, before a task has been assigned to the decision aid.
   - Trust in system capability for a specific task.
   - Trust in decision aid recommendation before it has been verified or implemented.
   - Trust in decision aid recommendation after it has been verified or implemented.
2. Completeness: Degree to which the user understands the conditions that affect trust at a temporal phase. Completeness is based on the grounds and warrant.
3. Resolution: Degree to which the user can reduce decision aid uncertainty by discriminating situations. More completeness leads to better resolution. Notice that this definition is different from the Lee and See definition of resolution.
4. Reliability: Amount and quality of data or information that underlies the trust assessment. Assessment can vary based on user experience. Reliability is based on the backing.
5. Calibration: Correspondence of trust to the true quality of decision aid performance. Calibration is based on the assessment of reliability.

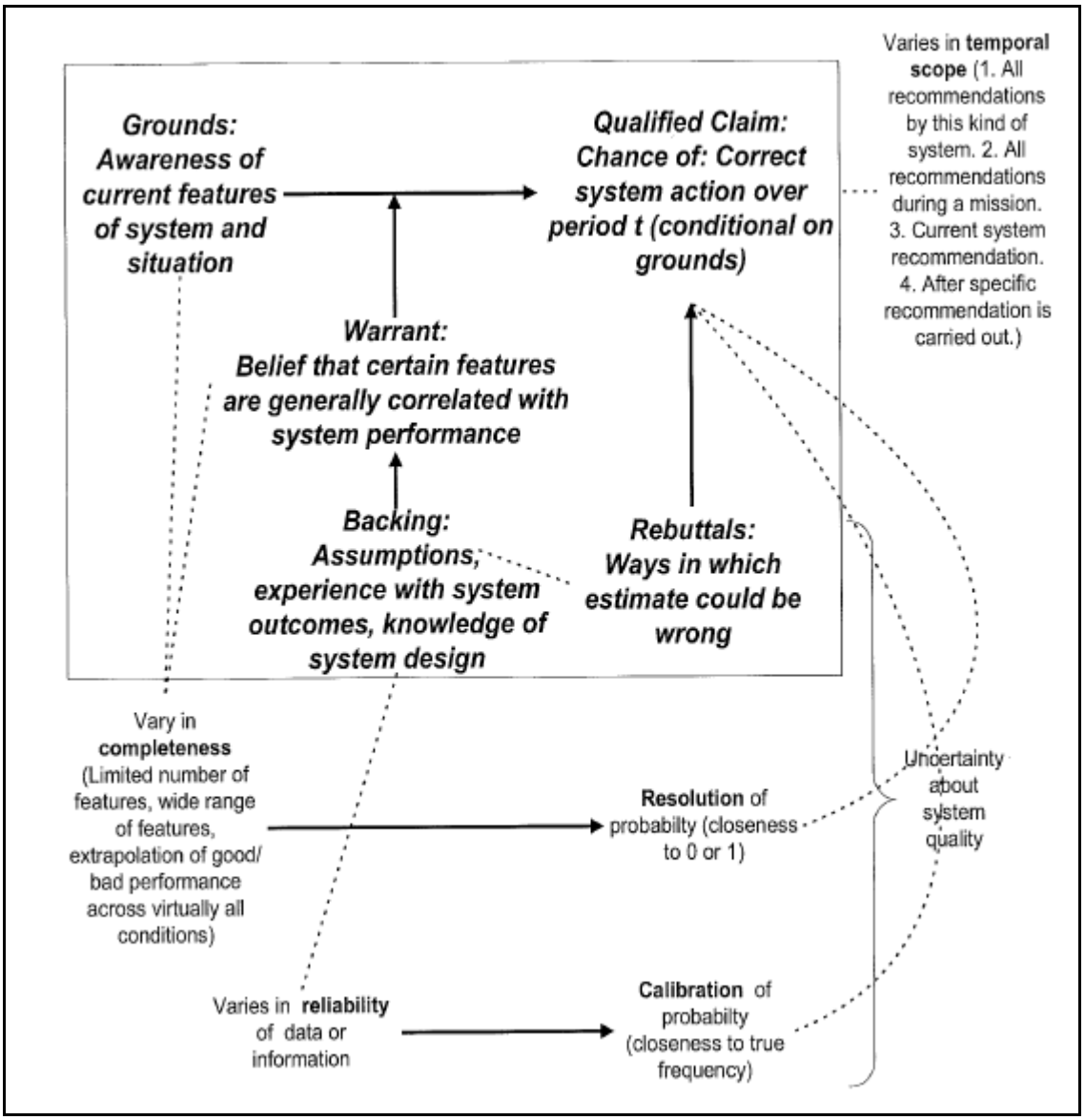


Figure 7: Argument-based Probabilistic Trust Model
(Cohen, Parasuraman, & Freeman 1998)

Event trees are used to form the quantitative model of trust. The purpose of the quantitative model is to explore the primary qualitative aspects of the APT model by clarifying the trust parameters and revealing their interactions. An event tree is used to visualize the acquisition of information required to update trust over time. The event tree includes all of the factors known to affect the aid's accuracy, presented in the order that the user is likely to observe them. Each path through the tree is a possible scenario that results in the successful or unsuccessful

contribution of the decision aid. The event tree is a summary of the effects on the user's mental model of the expected decision aid performance. Users with differing levels of experience, mental models, completeness of information, etc., will have different event trees and trust calculations. The numbers used in the assessment do not need to act like probabilities or correspond to the true relative frequencies of events. The assessment does not even need to be numbers, but can be qualitative assessments. The only requirement for resolution is that the tree must be able to discriminate between different situations. Calibration depends on the numerical assessment of trust. Miscalibration can be caused by incorrectly assessing probabilities. The purpose of calibration is not to calculate correct numbers to represent trust, but to "promote discriminations among situations that vary significantly in their implications for performance" (Cohen, Parasuraman, and Freeman 1998). For this reason Cohen, Parasuraman, and Freeman conclude that resolution is more important than calibration.

The APT model was designed to be used for developing training strategies for decision aid users. The authors do not advocate having decision makers learn about the APT model or event trees, or assess their trust with probabilities. Table 4 explains why the authors believe their APT model is superior to Muir's concept of trust:

Types of expectation (Barber)	Corresponding element of APT	Distinctions omitted by Muir's model
Persistence Physical Biological Social	Persistence represents a prior bias regarding the trustworthiness of very broad classes of systems (physical, biological, and social). Thus, it involves the longest possible *temporal scope* of trust judgments, and corresponds to a phase of decision aid use prior to knowing anything about a decision aid other than that it is a physical system used by a biological system within a social organization.	More differentiated trust assessments involve more limited temporal scope, and appear to have more relevance to decision aid use than highly generalized biases. For example, Muir's model omits trust in a particular aid over the span of its existence, trust in a particular aid during a particular type of mission or task, and trust in a specific aid conclusion.
Competence Skill-based Rule-based Knowledge-based	Judgments of competence simply mean that the type of task undertaken by a decision aid can form part of the *grounds* for assessing trust. For example, a particular system may have a better chance of successful performance in rule-based tasks than in knowledge-based tasks.	This is only one of many variables that can affect predictions of system performance. Far more differentiated judgments are possible. For example, a medical expert system might be better for diagnosing infectious diseases than pulmonary disorders; a planning aid might be less trustworthy when a particular factor, e.g., rotorwash, is important; and so on.
Fiduciary responsibility	Fiduciary responsibility involves making assumptions about the good motives of system designers. As such, it is a sort of *backing*, or source, for trust judgments. In the absence of more direct experience with an aid, users might have to fall back on such assumptions.	Fiduciary responsibility is only one sort of assumption users might make (e.g., they might assume the worst regarding the designers' motives or competence). In addition, assumptions are only one kind of backing for a trust assessment. Other sorts of backing include direct experience with the aid, talking with other users, analogies to other kinds of aids, and design knowledge.

Basis of expectation (Rempel et al.)	Corresponding element of APT	Distinctions omitted by Muir's model
Predictability (of acts)	Low to moderate completeness, high trust: The aid has been observed or is understood in a limited range of conditions and has been found to perform well.	As completeness increases, more and more conditions of performance are observed, but overall trust may either increase or decrease. The resolution of trust assessments, however, will increase as the user differentiates conditions of good and bad performance and makes more specialized assessments. Reliability and calibration may also increase with experience, independently of whether trust increases or decreases.
Dependability (of dispositions)	Moderate to high completeness, high trust. The aid has been observed or is understood in a wider range of potentially degrading conditions and has been found to perform well.	
Faith	Highest completeness, high trust. The aid has been observed or is understood in so many conditions and found to perform well that it is inferred / assumed to perform well everywhere.	

Table 4: Muir Framework compared to APT
(Cohen, Parasuraman, & Freeman 1998)

2.4 Ecological Interface Design (EID)

The display used in an automated decision aid affects both the user's understanding of internal algorithms and how the user controls the automation. The keyhole problem occurs when a user cannot find the information they need (Sheridan 2002). This problem arises when there are multiple menus, and the user forgets how to navigate various menus to find required information. Adaptive formatting allows the user or the automation to change the display depending on the situation or the user's personal preferences. However, frequently changing displays can make finding information even more difficult. EID is a method of display design that matches the automation controls and displays to the physical reality of the system and the user's mental model. The display is designed to show users how to interact appropriately with the system by embedding system constraints into the display. It also helps users to find patterns in time and space that would improve their decision making (Sheridan 2002).

EID uses an abstract hierarchy to integrate the goal relevant constraints operating on a system. According to Vicente and Rasmussen (1992), an abstract hierarchy has five levels:

1. Functional purpose: purpose of the system
2. Abstract function: causal structure of the process
3. Generalized function: basic function
4. Physical function: connections between components
5. Physical form: appearance and location of components

The higher levels of the hierarchy represent information about the system purpose, while the lower levels represent elemental data about the physical implementation. The hierarchy is goal oriented, which means a goal at a higher level can be broken down into constraints needed to attain that goal at a lower level. Wickens and Hollands (2000) explain that for a variable at a given state, variables beneath it in the hierarchy explain 'how' the given state is changed while variables above explain 'why' the variable is changed. Figure 8 illustrates this point. The hierarchy is used to represent the work domain by providing users with the information necessary to cope with unanticipated events. The display must be able to transition from level to level, depending on the needs of the user.

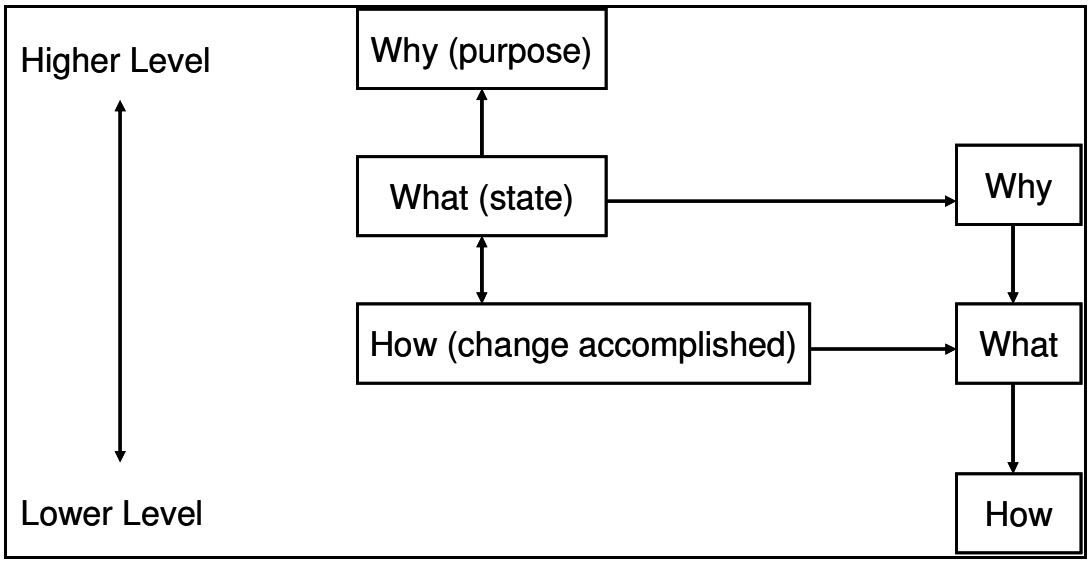


Figure 8: Wickens and Hollands EID Hierarchy
"Schematic representation of abstraction hierarchy. The column on the right represents a shift of attention to a lower level in the hierarchy." (Wickens and Hollands 2000)

EID is built around the way humans perceive information, which is pointed out by Lee and See's (2004) requirement to foster appropriate trust by presenting information in a manner that is compatible with the analogic, analytical, and affective cognitive processes. Vicente et. al. (1995) describe the goals of EID: "to allow operators to effectively exploit their powerful perception and action capabilities, while also providing the support required for problem solving activities."

Rasmussen (1999) adds to the principles by stating that intention information must be provided so that users understand why the system does what it does. The display should provide situation dependent affordances that allow the user to act on the system differently given the constraints on the situation. The display should also force the user to adopt an accurate mental model. The proper implementation of EID makes the "black box" of automation more transparent to the user, improving the user's ability to interact with the automation.

[This Page Intentionally Left Blank]

Chapter 3
Trust-Based Design of Human-Guided Algorithms

This chapter combines the strengths of the various models and theories discussed in Chapter 2 to develop an automation design approach that incorporates trust.

3.1 Building Trust in a Human-Guided Algorithm

A model of trust in decision aids, based on the Lee and See model, is presented in Figure 9 to further define the aspects of automation that affect human trust. The Lee and See (2004) model (see Figure 5) conceptually reveals the dynamic relationships between human trust and the use of automation. The modifications identify the methods for impacting human trust.

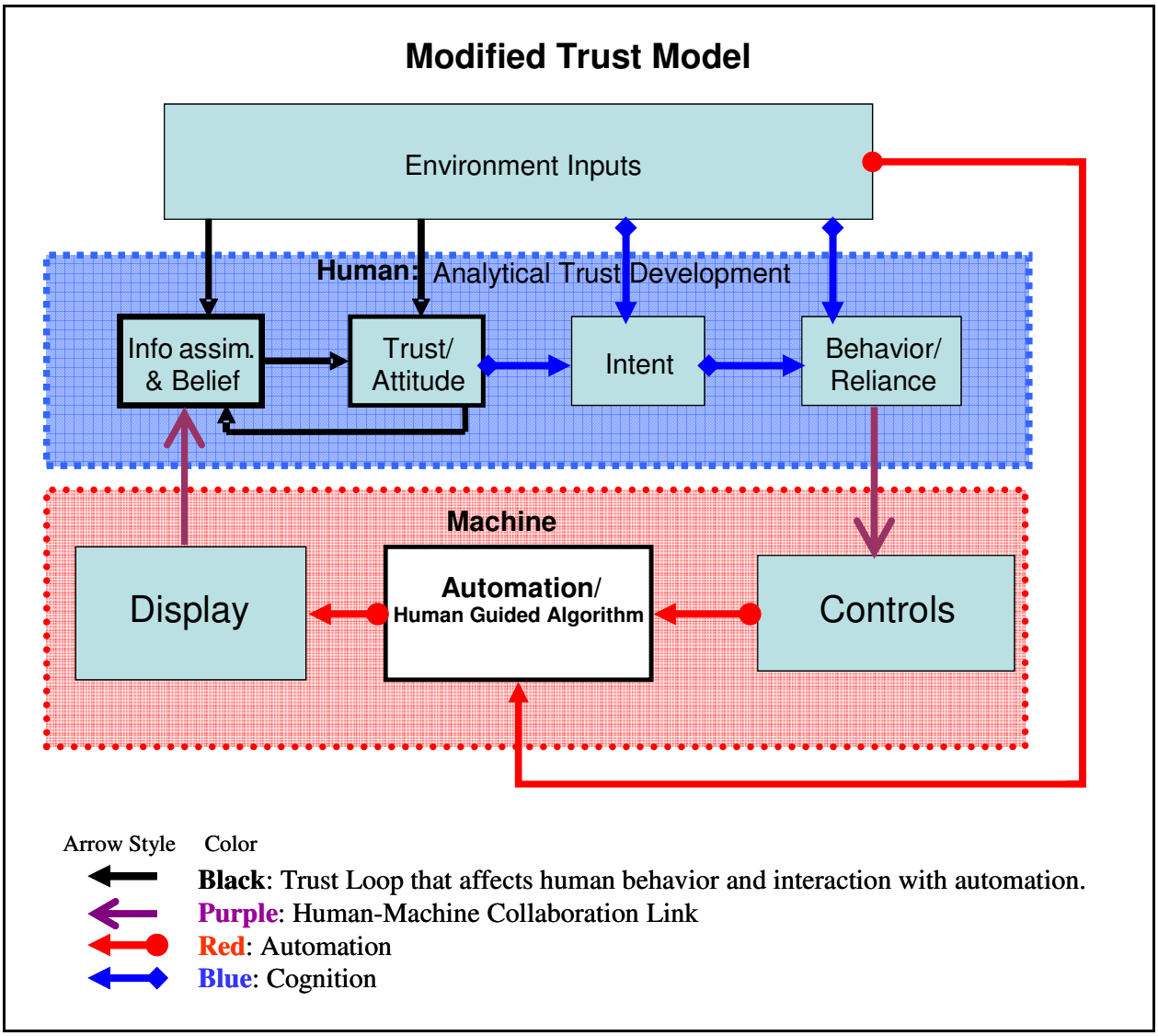


Figure 9: Modified Lee and See Trust Model

The first major modification is the addition of the Controls box. After the Behavior/Reliance decision is established, human interaction with the automation depends on how well the human can communicate his/her intentions to the automation. The Controls determine how the user can influence automation or steer HGAs, therefore having a direct effect on the outcome of the automation. Trusting the human-machine collaborative components requires the user to evaluate his/her ability to interact with the computer (the controls) and the computer's ability to interact with the user (the display).

The second modification involves separating the model into Human and Machine components. The Human components, defined by Lee and See, include the cognitive aspects

related to trust: Information Assimilation & Belief, Trust, Intent, and Behavior/Reliance. While these Human components are influenced by the automation and environment, they occur entirely through human cognition. The Machine components of Display, Automation, and Controls depend on the designers and design process.

The two arrows connecting Behavior/Reliance to Controls and Display to Information Assimilation/Belief show the process of Human-Machine collaboration. How well the user can interpret and interact with the display might be the limiting factor to his/her Information Assimilation & Belief. Similarly, how well the user can control the automation might be the limiting factor to automation performance. To optimize automation performance it is necessary for both the controls and display to support human-machine interaction for all possible environmental conditions and scenarios. The proper design of controls and display improves system transparency and user understanding, therefore helping to foster appropriate trust and behavior/reliance based on the abilities of the HGA. Since the design of controls and display impact the dynamic trust relationships, the controls and display should be designed to improve the user's calibration of appropriate trust.

A key component of designing an HGA is to reveal how the options for manipulating the parameters through the controls will affect the resulting solution. An approach, applying ecological interface design (EID) to the design of the display and controls, is presented as a method for ensuring appropriate transparency of the HGA. The traditional theory of EID, states that the constraints of the physical system should be revealed to the user through the user interface (Rasmussen 1999). By extending EID to HGAs, the constraints of the algorithm are also revealed to the user, allowing them to better understand the black box automation.

For proper trust calibration, the user must be able to evaluate trust in relation to Lee and See's (2004) three dimensions of trust (purpose, process, and performance). First, the operator must evaluate how the HGA meets its intended purpose of design, the purpose dimension of trust. Since the purpose of an HGA is to solve an operator defined problem, the operator must be able to evaluate how well this is achieved. Through the use of controls and display the operator builds a mental model of how the algorithm solves the problem. The user needs an accurate mental model to operate the HGA according to its intended design and purpose. Next, the operator must evaluate the process dimension of trust. He must evaluate the HGA's internal models, solvers, search algorithms, calculations, and data. These processes must be analyzed

with respect to the user's goals, the environment, and the problem being solved. Finally, the user must be able to evaluate HGA performance. This includes historic HGA behavior and performance and current performance while using the HGA. The operator must be able to evaluate his own qualifications as a decision maker, understanding of and ability to operate the HGA, and performance while operating the HGA. The users must be supplied with an appropriate amount of information during the solution process to allow him/her to trust the HGA and final solution appropriately. To calibrate trust, the purpose, process, and performance dimensions must be addressed in the design of the HGA.

The purpose of trust-based design (TBD) is to address the three trust dimensions by informing the user how to trust the individual components of the HGA at specific times. Through the design of human-machine collaborative controls and displays, and providing the user with trust information, TBD can address the issue of trust. TBD should be considered during development of the HGA because it can directly affect the solution techniques and final solution quality.

It is necessary to note that the purpose of TBD is not to blindly increase trust in any component of the HGA. Increasing user trust without due cause is irresponsible and can lead to poor decisions and accidents. Trust in the computer and its processes should be accurately represented depending on the situation and ability of the computer. Each user or team of users should trust themselves differently depending on the situation and their own abilities. TBD should help an operator calibrate their trust, avoiding both over-trust and mistrust. Mistrust leads to incorrect use, which results in the automation not being trusted or used at all. TBD should only be used to accurately represent trust in all necessary components of an HGA at all times of the solution process, including the strengths and vulnerabilities of the final solution. The objective of TBD is to represent the three dimensions of trust accurately in all necessary components of an HGA.

3.2 TBD of Controls

The HGA should be designed around the users and the controls that allow them to interact with the algorithm. For example, an HGA that allows the user to change the numerical search strategy will be different from an HGA that has a fixed search strategy but allows the user to focus the search by improving a specific area of the problem. Both HGAs may solve the same

problem, but the HGA design, solution process, solution quality, and controls will be different. For this reason, it is important to determine how the HGA will be controlled before the algorithms are built. This design decision should be based on the users of the system and their decision-making process.

The types of controls to consider can be categorized by Shahroudi's (1997) methods of solutions steering:

1. Direct Control on the Search Progress

   Controls allow the user to:

   - Select a starting point and direction for the search.
   - Continually adjust the search direction and location.
   - Focus the search in a specific area of the problem.

2. Direct/Indirect Control on the Search Problem

   Controls allow the user to:

   - Change the problem definition: objective function, constraints, parameters, variables, models, assumptions, etc.

3. Direct Control on the Numerical Search Strategy

   Controls allow the user to:

   - Select among various search algorithms: local, global, probabilistic, deterministic, greedy, various heuristics, etc.
   - Tune the algorithm.

HGA design can encompass multiple methods of solution steering. The specific solution steering methods to include depend on the individual problems being solved by the HGA, the capability of the users, and the types of solutions the HGA user is trying to find. As with any type of automation, adding more capabilities to a system might have unintended drawbacks. Providing a user with too much control over the HGA's solution process can be overwhelming or confusing. A task analysis and function allocation should be performed to determine how adding controllability to an HGA will alter the solution process and the quality of the solutions

generated. The HGA design should consider the complimentary strengths of humans and computers and the evaluative criteria proposed by Parasuraman, Sheridan, and Wickens (2000). Once it is determined that a specified set of controls will allow a user to find high quality solutions to the given problem, the algorithm can be built to accommodate the chosen solution steering controls.

The design of the controls should attempt to provide three pieces of information related to the three trust dimensions:

1. Purpose - How to use the controls correctly.
2. Process - High level indication of how the controls change the HGA process.
3. Performance - What is likely to happen when specific control inputs are made.

Controls that match the designers' intended *purpose* to their actual use must be intuitive to operate by revealing how to use them correctly. Showing how the controls change the HGA *process* provides the user with a high level of understanding necessary to allow him/her to steer the HGA process. With respect to *performance*, controls that predict how specific control inputs affect the system provide the user with a detailed understanding of how his inputs can change the system in the current situation. Providing performance information should help the user better control the system, but may come at a cost. For example, the sensitivity analysis required to predict specific future control inputs might require a significant increase in computational time. With ecological interface design (EID), the overall design of the controls should support the controllability of the system by improving user understanding with appropriate system transparency. Both the type of control and its design are crucial to this goal.

3.3 TBD of Displays

The foundation of HGA displays rests on using the principles of EID to reveal the inner workings of the algorithm. The expertise of the user is likely to be in the problem domain, which allows the user to steer the algorithm intelligently to solutions that the user believes are of high quality. The user is unlikely to understand or know the logic of the algorithm and why/how the algorithm makes the decisions that it does. For this reason, the display must provide the necessary information for the user to refine his/her mental model of the logic applied by the

algorithm and thus guide the algorithm toward a solution that meets his/her objectives. Since the problem, user, and environment usually change, the display must be built to accomplish these objectives in various situations. This can be accomplished by a robust or adaptable design. Displays that answer the following questions might support the user's search for a high quality solution:

- What direction is the current search heading?
  - Display: How the algorithm is attempting to improve the current solution.
- Where is the current search now?
  - Display: Current progress of the algorithm.
- Where has the search been?
  - Display: What parts of the solution space have already been explored.
- How did the current search arrive at where it is now?
  - Display: What control inputs directed the algorithm to the current solution.
- Why did the current search arrive at where it is now?
  - Display: Why past control inputs caused the algorithm to go to the current solution.
- How can the current search move somewhere else?
  - Display: What modifications to the control inputs need to be made to cause specific improvements to the solution or to explore other areas of the solution space.

The display must provide the user with an accurate mental model of the internal workings of the HGA. The type of information, when it is displayed, and how it is displayed can provide the user with high trust resolution and specificity during the solution steering process. Appropriate levels of transparency allow the user to trust various components of the HGA correctly during specific situations. The user must understand what information is relevant to the current situation and his/her goals as a decision maker. Necessary information includes anything operators need to operate the system, evaluate HGA performance, and generate high quality solutions, as quickly and efficiently as possible. This information, trust information, should be provided to improve the user's calibration of trust.

The principles of EID should be used to reveal the algorithmic constraints to the user. The display must provide the user with the necessary level of understanding of the past and present HGA behavior and performance, so that he/she can steer the algorithm in the preferred direction. The display must allow the user to observe and evaluate how the HGA processes' (available algorithms and heuristics, decision-making tools, steering controls, etc.) affect the solution process. Since the performance of certain HGA processes are problem dependent, the user must know how the processes are changing the search and if they are successful for the current problem. In addition, information about algorithm decision rationale, availability of other solution steering methods or strategies, etc., can improve understanding and trust of the HGA. After deciding what information needs to be displayed, how it is displayed affects how the user interprets it. The display is the critical link that makes the interaction between the algorithm and the human possible. A display that conveys the necessary constraints and relationships of the algorithm makes it possible for the human to understand and control the HGA, facilitate human-machine collaboration, and efficiently generate high quality solutions.

During the operation of the HGA search, the user will generate multiple solutions. The user might want to compare various solutions as a means of evaluating the performance of the HGA or to select a final solution from a collection of possible solutions. The display should allow the user to compare multiple plans with many metrics and confidently select the plan that outperforms the rest. The exact display design depends on the types of plans being generated and the metrics that interest the user. The user should also be able to peruse previously generated solutions. If the user does not like the direction the search is going, the user must be able to retrace his steps by directing the computer to return to a previous solution and begin the search again with a new search direction or technique.

3.4 TBD of Trust Information

Trust information includes anything that helps decision makers calibrate their trust appropriately for decision-making tools, their own abilities, and the situation. With respect to HGAs, this can include information about the reliability of data, models, calculations, user performance metrics, historic performance, etc. This additional information is used to supplement operator understanding that is obtained through the controls and displays, to provide necessary

information about the three dimensions of trust. The type of information that is required can be derived from the five trust parameters and by applying the abstract hierarchy approach.

3.4.1 The Five Trust Parameters

As explained in Chapter 2, Cohen, Parasuraman, and Freeman's (1998) Argument-based Probabilistic Trust (APT) model identifies five parameters necessary to assess trust: temporal scope, completeness, resolution, reliability, and calibration. In addition to the three dimensions of trust, trust information provided to decision makers must address these parameters.

Information should be provided throughout the solution process and must specifically address the four phases of temporal scope: before the HGA is used, when a task is assigned to the HGA, when the HGA solution is generated, and after the HGA solution is verified or implemented. As an alternative to providing trust information to the user during all temporal phases, some information can be provided during initial training.

During each temporal phase, completeness can be accomplished by providing the user with the correct amount and type of trust information at appropriate times in the HGA solution process. The purpose of providing complete information is to provide resolution to the decision maker. The APT definition of resolution is the degree to which the user can reduce HGA uncertainty during unique situations. An approach to providing plan metric reliability information to reduce uncertainty for a UAV mission planning HGA is presented in Chapter 4. Reliability is the amount and quality of the data used to provide trust information. Trust information must be reliable or trust calibration will be poor. Calibration is the matching of HGA user trust to the HGA's true abilities. Calibration quality can only be as good as the reliability of the trust information.

Cohen, Parasuraman, and Freeman (1998) use event trees to quantitatively model and calculate the trustworthiness of a system. They warn that the purpose of this calculation is to shed light on the five qualitative trust parameters and not to present this information to decision makers. They explain that the purpose of trust calibration is not to calculate correct numbers, but to provide resolution to the decision maker. However, other researchers regard calibration as the most important aspect of trust. It is the author's opinion that both calibration and resolution must be emphasized with trust information. Calibration allows the user to evaluate how much to trust the HGA and final solution by matching the user's trust to the HGA's true capability. Resolution

allows the user to discriminate between various decision options with differing trust implications, allowing the user to take appropriate action that improves the trustworthiness of the decisions.

The purpose of providing trust information is to calibrate user trust and improve the resolution necessary to generate appropriately trusted solutions, and in some cases produce solutions that are more trustworthy. The user's calibration must allow him/her to appropriately trust the HGA's individual components during specific times in the solution process. With proper calibration and resolution, the user should be able to avoid untrustworthy components and rely on trustworthy components in order to improve trust in the HGA and final solution. This can be accomplished by providing a calculation and analysis of the reliability of the HGA to help the user take appropriate action. For example, if there is high variability in the data that is used to define the problem or one of the models used in the calculation, then the solution will be of a lower quality. Providing the user with information on the quality and timeliness of the data and models may allow him/her to consider more accurate metrics in steering the HGA and deciding on a final solution. In cases when it is not possible to improve the trustworthiness of a decision, the trust information allows the user to assess the uncertainty and vulnerability of his decisions and the final solution.

3.4.2 Abstract Hierarchy

The first step to identifying the information needed for operators to calibrate their trust appropriately is to elucidate the complex relationships within the HGA. This can be done with a framework that breaks down the components of the HGA and links the relationships between them. One method of accomplishing this is with an abstract hierarchy, used by Vicente and Rasmussen (1992) for EID. An abstract hierarchy, built by the HGA design team, can show the connections between input data, algorithmic calculations, human input, and the final solution. The levels for an HGA abstract hierarchy may look like the following:

1. Functional Purpose: To develop high quality solutions that can be appropriately trusted.
2. Abstract Function: Methods of human-machine collaboration.
3. General Function: HGA solution methods - search algorithms, solvers, models, calculations, displays and control.

4. Physical System State: Current process and progress of the HGA, data, user experience and ability, HGA historical data.
5. Physical Model: Physical layout of system and components, computer code structure, computer hardware and software capabilities.

For the user to develop appropriate trust in the HGA and final solution, he/she must have an understanding of these connections between the levels of the hierarchy. The hierarchy reveals these connections, but does not provide numerical calculations to show the magnitude of the impact of one variable on other variables.

The abstract hierarchy also outlines the general order of importance to develop an HGA, starting with the functional purpose, and ending with the physical model. The lower levels should not be designed without considering the requirement of higher levels. First, the functional purpose defines the requirements of a high quality solution to the problem. Next, the abstract function reveals how human-computer interaction can be used to develop quality plans and the tools required for interaction and solution steering. The general function determines the specific models, calculations, and search methods that the algorithm is capable of performing. The physical system state describes the data, assumptions, user ability and experience, HGA historical performance, and other initial conditions that affect the quality of the solution from the start of the solution process. It also includes the current progress of the algorithm during HGA use. Finally, the physical model includes the physical layout and connections between the computer, display, controls, and user. The physical model should be the final design consideration, but if done incorrectly it can drastically hinder the quality of the final solution. For example, inefficient computer coding or an inadequate computer processor can lengthen the time it takes to perform certain calculations, which can affect the quality of the final solution in time-critical settings. In some cases, the physical tools used to steer the HGA process should be a design consideration: joystick, mouse, keyboard, touch screen, multi-screen display, etc. Like many design processes, as lower levels of the hierarchy are fleshed out, higher levels may need to be adjusted, leading to an iterative process.

3.4.3 Calculations

To quantify and compare trust among various components and their interactions, numerical calculations must be performed. The type of calculations performed depend upon the analysis used to evaluate trust in the HGA. The purpose of numeric trust calculation is to provide a magnitude to accompany the relationships shown on an abstract hierarchy. Since numerical calculation of trust may not be possible, feasible, or beneficial, additional research is required to determine methods for trust calculation. Numerical values will not be appropriate in all situations, and should only be used when they provide accurate and interpretable information to an HGA user. The purpose of the following discussion is to highlight some of the considerations necessary for numeric trust calculation.

The first step is to determine what type of numeric trust information would benefit the HGA user. For example, in some cases it might be sufficient to provide the user with information about the variance or reliability of the data. In other cases, it might be necessary to explore how data instability affects the models and metrics calculated by the algorithm. Depending on the required information, there must be methods of assessing reliability in some of the individual components defined on the abstract hierarchy: data, models, search algorithms and heuristics, user performance and experience, collaborative methods, etc. In some cases, humans with expert knowledge on a specific component of the HGA might be better at evaluating trust in that component than numerical calculation performed on a computer. The computer might even be able to evaluate the human's performance and provide recommendations for improvement. Overall, it must be determined what trust information is necessary, attainable, quantifiable, and how it can be employed to compliment the abstract hierarchy.

The next step is to quantifying the interactions between the HGA components. Once the reliability has been assessed, it must be determined how the components interact to either decrease or increase reliability in the system. Examples of possible methods for quantifying these interactions include probability analysis, statistics, event trees, historical data, decision analysis, and simulation. The methods used are problem dependent and require careful consideration to determine their utility to the decision maker.

Any numerical value must be linked to the correct attitude of trust and vice versa. If an expert rates a component of HGA performance, his attitude towards trust must be quantified if it is to be used to calculate interactions with other components. One method of quantifying trust is

to provide the users with rating scales that correspond to numerical values, similar to Likert scales. After numerical calculations are performed, these trust values must be matched with an appropriate human attitude towards trust. Another method of calibrating trust values to how they should be interpreted is to evaluate how solutions with specific trust values historically perform in simulations or preferably in real-world implementation.

The final considerations are how to present the numeric trust information to a user, how the user should interact with it, and how the user can take action to make the solution more robust against the identified uncertainty. TBD should be used to address the display and control of trust information and to consider additional methods of human-machine collaboration that can make solutions more robust.

3.4.4 Drawbacks and Side Effects

The drawbacks of this approach must be considered to determine if providing trust information is feasible, reliable, necessary, and beneficial to HGA users. The following is a list of topics that must be weighed against the possible benefits of trust information:

- Adding levels of automation and collaboration to existing automation can have negative consequences, such as increased or altered workload. Before any trust information is provided to an HGA user, a functional analysis should be performed to determine how this information would affect the HGA process.
- The calculations, calibration, and interpretation of numerical trust values can be complicated and time consuming. They might not be feasible in time-critical situations or necessary in low-risk scenarios.
- How can the trust information be trusted?

The ideas pertaining to the development of an HGA abstract hierarchy and the calculation and calibration of accurate trust information must be fleshed out for a given problem before they can be successfully implemented.

[This Page Intentionally Left Blank]

Chapter 4
Human-Guided Algorithm for Vehicle Routing

4.1 Problem Overview

Forest, et al (2007) developed a human-guided algorithm (HGA) as a test platform for experiments on Human Machine Collaborative Decision Making (HMCDM). The HGA allows an operator to develop a plan for routing unmanned aerial vehicles (UAV) to achieve a set of objectives. The UAVs take off from their home base and travel separate routes to prosecute enemy ground targets. Ground targets are assigned numerical values depending on their importance. Each target requires a certain number of weapons to be destroyed and each UAV is constrained to carry a limited amount of munitions. In addition, some targets represent surface to air missiles (SAM) that can destroy the UAV. The goal of the human operator is to search the feasible solution space for a routing plan that accrues as much value as possible by destroying enemy targets. The operator must balance this goal with the amount of time it takes the UAVs to complete the mission, the amount of fuel used by the UAVs, and the risk of the UAVs being destroyed. The models used in the algorithm simplify real-world UAV operations and mission planning, but they are sufficient to test human-machine interaction.

4.2 Algorithm Design

The algorithm uses a "cluster first, route second" heuristic to generate an initial plan. Clusters are generated heuristically by considering target locations and requiring that one UAV has enough ordnance to destroy all of the targets. Because it is possible that there are not enough UAVs to strike all of the generated clusters, the clusters with the highest value (sum of the target values) are assigned to UAVs. The routing step uses a farthest insertion heuristic. For each cluster, it starts by creating an edge between the home base and the farthest target. It then finds

the target that maximizes the minimum distance to any of the nodes already in the existing route. Next, it inserts this target into the route using a cheapest insertion heuristic that minimizes the distance traveled by the UAV. It does this until all of the targets are inserted.

The initial plan is then assigned a score depending on the weighted sum of five metrics:

1. Total Value: Total value of targets struck.
2. Fuel: Total fuel used by all UAVs; the amount of fuel used is determined by the distance traveled by the UAV - assuming a constant burn rate. UAVs travel in a straight-line distance from target to target.
3. Time: Total mission time.
4. Expected Attrition: Expected number of UAVs destroyed calculated by the attrition probabilities of each UAV. A UAV only accrues risk when it flies through a SAM site's threat radius and is not prosecuting the threatening target. It is assumed that when a UAV is prosecuting a threatening SAM, the UAV's jamming capability allows it to accrue an insignificant amount of risk.
5. Utilization: Total number of weapons used divided by total weapons available.

The scores of the metrics are standardized from 0 to 1. A weighted sum of fuel, time, and expected attrition is subtracted from the weighted sum of total value and utilization to create the overall plan score in Equation 5. The weights used to score the overall plan are selected by the user. The objective function maximizes total value and utilization and minimizes fuel, time, and expected attrition. The objective function is self-explanatory:

```
PlanScore =      WeightValue * (TotalValue / TotalAchievableValue)
                +WeightUtilization * (TotalWeaponsUsed / TotalWeaponsAvailable)
                - WeightDistance * (TotalDistance / DistanceLimit)
                - WeightTime * (TotalTime / TimeLimit)
                - WeightAttrition * (ExpectedUAVLost / NumberUAVs).
```

Equation 5: UAV Routing HGA Objective Function

It is important to note that the standardizing values (TotalAchievableValue, TotalWeaponsAvailable, DistanceLimit, TimeLimit, NumberUAVs) are problem dependent and not set by the user.

After the initial plan is generated, a large-scale local neighborhood search begins. First, the algorithm determines all of the possible plans that can be generated by making one feasible change to the current plan. This set of plans is called the neighborhood. The possible plan changes, called moves, include:

- Edge Swap: An edge from a route is swapped with an edge from another route.
- Target Swap: A target from a route is swapped with a target from another route.
- Insert Target: An unassigned target is added to a route.
- New Route: A new route is created with two unassigned targets and the base.
- Remove Target: A target is removed from a route.
- Replace Target: A target is removed and an unassigned target is inserted into the route.
- Reverse Route: The direction of the route is reversed, which changes the time that the UAV prosecutes specific targets and can change the amount of risk accrued by UAVs. For example, prosecuting a SAM earlier in the mission removes it as a threat to passing UAVs later in the mission.
- Move Target: An assigned target is moved from its route and inserted into another route using the distance based cheapest insertion heuristic.

The moves that require inserting a target into a route, Target Swap, Insert Target, Replace Target, and Move Target, use a cheapest insertion heuristic that is based on distance. Because the neighborhood does not include every possible plan, it is considered a local search. The score of each plan in the neighborhood is calculated and the plan with the best score becomes the new plan. From here, a new neighborhood is generated, and a new plan is selected to maximize the score. The search continues until a plan is reached that has no neighbors with an improved score. This solution is the locally optimal solution to the routing problem, and is displayed to the user.

4.3 Human-Machine Collaboration Design

To start the algorithm the user must define a set of coefficients (or weights) that define how much the objective function weights the metrics used to calculate the plan score. The algorithm will then begin the above process and display to the user the plan generated after the "cluster first, route second" heuristic and search. The user can then change the coefficients and perform the local neighborhood search again. The local neighborhood search begins at the current plan and looks for improving moves based on the new coefficients until a plan is reached with no improving moves. This process of changing the coefficients and searching the solution space continues until the user is satisfied with a plan.

A user might wish to change the coefficients for various reasons. If the plan does not match his objectives, he can change the coefficients to weight some metrics more than others. The list below shows how increasing one coefficient from its previous value will likely change the next locally optimal plan:

Increase value weight	→ Increase total value metric
Increase fuel weight	→ Decrease distance metric
Increase risk weight	→ Decrease attrition probability metric
Increase utilization weight	→ Increase utilization metric
Increase time weight	→ Decrease time metric

The new plan that is generated will reflect the changes made to the coefficients. It is possible that if the coefficients are not changed enough between searches, the current plan will still be locally optimal. The user might wish to change the coefficients dramatically to get out of the neighborhood and then begin the search again to explore a new area of the solution space. By changing the coefficients, the user can explore the solution space and generate plans that meet his objectives. This HGA uses Shahroudi's (1997) solution steering methods of indirect and direct control on the search problem. It also allows the user to decide when the cost of future searches outweighs the benefits of making improvements to the solution.

Due to the nature of maximizing a linear objective function, the relative value of the coefficients to each other, and not their specific values, determines the direction of the local neighborhood search. For example, if all of the coefficients are increased by some factor, the

score increases by that same factor along with the scores of all of the neighborhood plans. Consequently, the current plan is still locally optimal. Therefore, the HGA is actually controlled by the user changing the ratio of the coefficients.

The HGA also allows the user to make manual changes to the current plan, and then begin the local neighborhood search from that new plan. The user can create and delete routes, change the ordering of targets on a route, and add or remove targets from a route. The user can make as many changes as necessary, which may create a new plan that is outside the old plan's local neighborhood. Making one manual change that is equivalent to a move in the local neighborhood search decreases its score, because all neighborhood plans have a lower score. However, making a different type of move or more than one move can create higher scored plans. In a previous experiment conducted with this HGA test platform, the users preferred adjusting the weights to making manual modifications to the plan (Forest, et al. 2007).

4.4 Initial HGA Interface

As shown in Figure 10, the initial graphical user interface (GUI) contains six components: Sliding Bars control, Manual Modification control, Local Search button, Problem and Route Information Bar, Plan Metrics Box, and the Plan Graphic display. The Sliding Bars control contains five sliding bars that allow the user to set weights for the five metrics in the objective function: value, fuel (distance), risk, utilization, and time. Each bar can be set with the mouse to any integer between 0 and 100 inclusive. The Manual Modification control is below the problem definition control. The user can click on radio buttons and then on route legs or targets to make manual modifications to the plan.

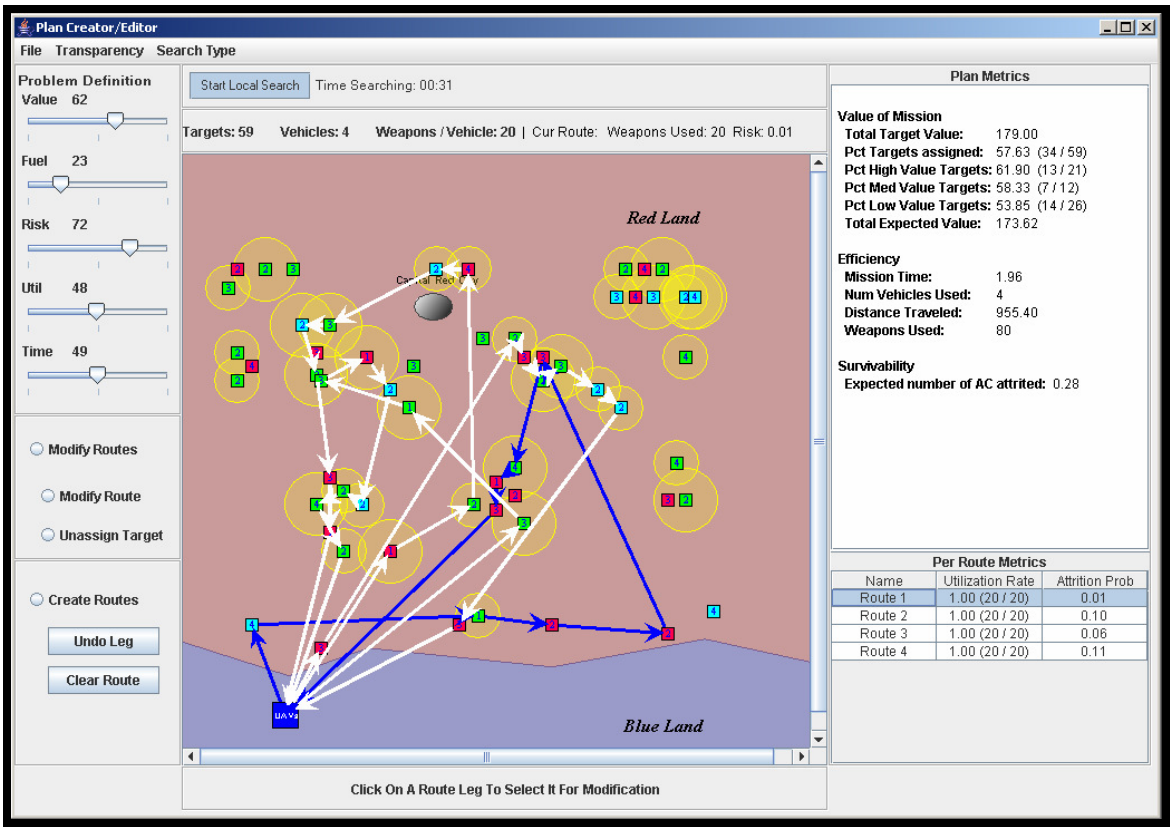


Figure 10: Initial HGA Vehicle Routing Interface – Level 0
(Forest, et. al 2005)

The Local Search button begins the search when the user selects it with the mouse. The time it takes to complete the search is displayed next to the button. The user knows the algorithm is still searching when the time continues to increment. The Plan Metrics Box lists relevant plan metrics. Underneath the metrics is information about individual routes: utilization rate and attrition probability. The Problem and Route Information Bar is below the Local Search button. It displays the number of targets and vehicles in the scenario and the number of weapons that each vehicle carries. The text turns red if an infeasible solution is generated. Infeasible solutions can be generated by manually creating more routes than available UAVs or manually directing a UAV to use more munitions than available. If a route is selected with the mouse, it displays the number of weapons used on that route and the UAV's probability of attrition.

The Plan Graphic display shows the UAV home base, targets, threat rings, UAV routes, landmarks, and terrain features. Figure 11 provides a key for these items. High valued targets are worth 10 points and are indicated by a red box. Medium valued targets are worth 5 points and are indicated by a blue box. Low valued targets are worth 1 point and are indicated by a

green box. The number in each box represents how many munitions are required to destroy the target. There are two types of threat radii created by SAM sites; small yellow circles represent a small threat radius and large yellow circles represent a large threat radius. The intensity of the threat is the same; however, a UAV is likely to remain in a large threat radius for more time than a small threat radius and therefore accrue a higher attrition probability. After a SAM site is prosecuted, it discontinues to pose a threat. The thickness of the UAV route edges indicates the UAV's attrition probability on that edge. A thicker edge indicates a higher attrition probability than a thinner edge.

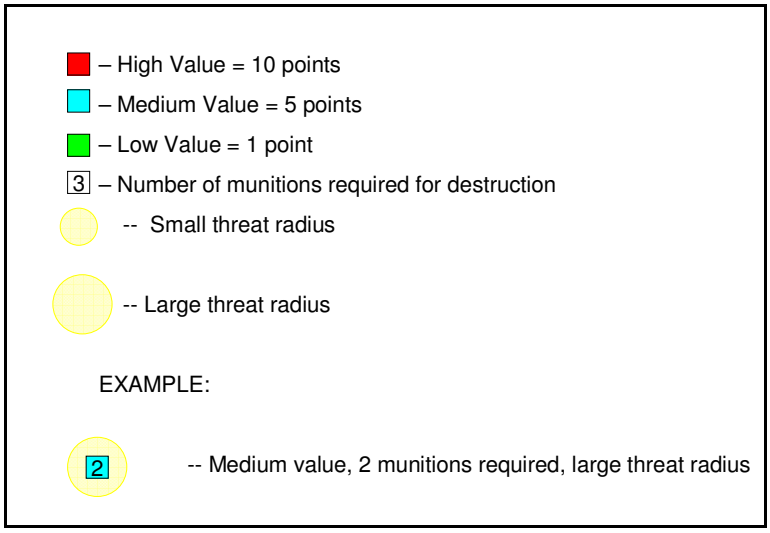


Figure 11: Key to Plan Graphic Display

The user can select the "File" menu and then "Get Plan Info" to compare up to four plans at a time in the Plan Comparison window (see Figure 12). All of the plans discovered by the user are labeled in numerical order and saved in a file that can later be accessed. The user can load one of these plans into the Plan Comparison window or into the Plan Graphic display. If loaded into the Plan Graphic display, the next search performed by the user will begin from that plan. This option allows the user to go back to a previously discovered plan and perform new searches from it.

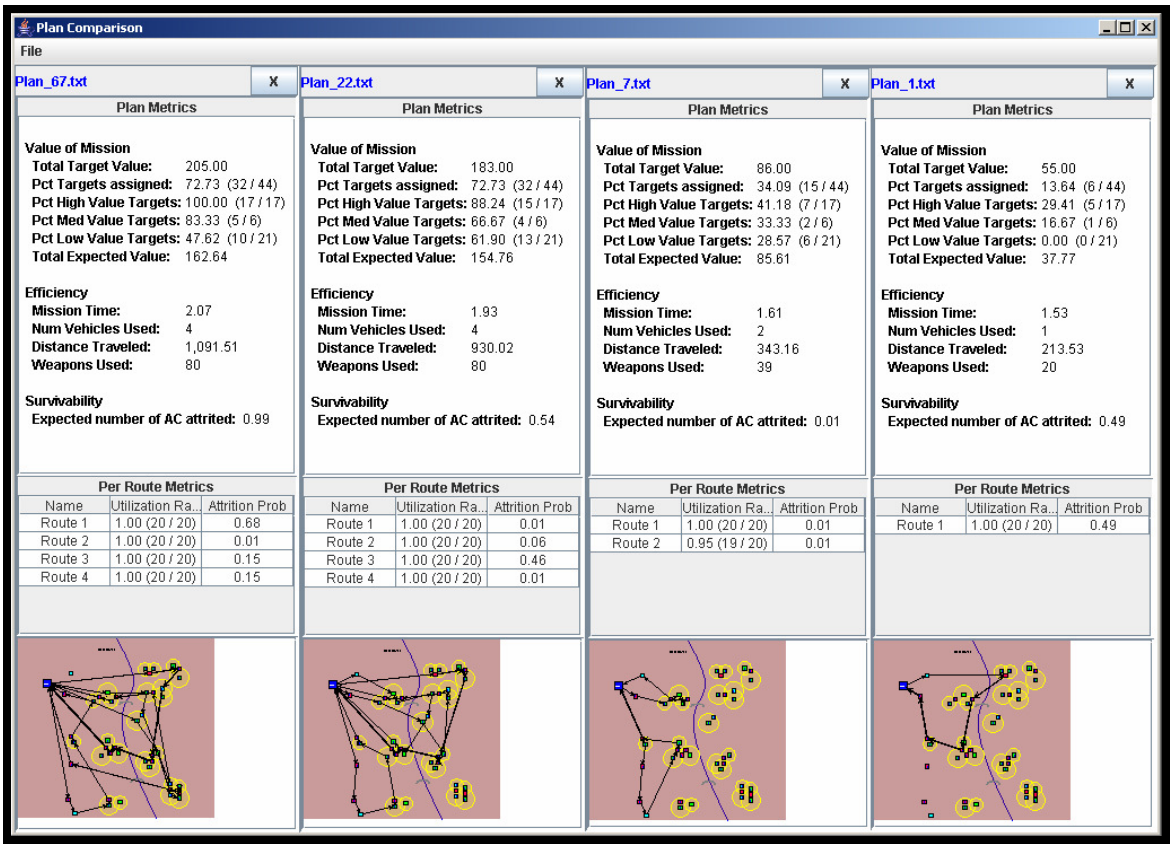


Figure 12: Plan Comparison Window

4.5 Applying Trust-Based Design of Controls and Display

The trust-based design (TBD) principles and approaches presented in this thesis were applied to refine and improve the UAV vehicle routing HGA described above. Previous experiments with the initial HGA design revealed difficulty in understanding the relationship between the objective function coefficient inputs with the Sliding Bar control and the resulting solution, thus reducing operator trust in the HGA. In addition, operators had difficulty understanding the localness of each search and visualizing their progress as they searched the solution space. To improve understanding of the HGA and calibrate appropriate trust, the following list of eleven goals, summarized from Chapter 3, was used to design TBD tools:

1. Show how to use the controls correctly.
2. Show how the controls change the HGA process.
3. Show what is likely to happen when specific control inputs are made.
4. Improve the user's mental model of the solution process.

5. Reveal appropriate system transparency at specific times and situations.
6. Display how the algorithm is attempting to improve the current solution.
7. Display the current progress of the algorithm.
8. Display the solution space that has already been explored.
9. Display the control inputs that yielded the current solution.
10. Display *why* those control inputs caused the algorithm to go to the current solution.
11. Display what control inputs yield specific improvements to the solution or explore other areas of the solution space.

The TBD analysis led to the following modifications to the HGA: one control and five displays. Two of the tools, the Rationale Window and Plan Cycle Option were added to the HGA GUI. The Rationale Window displays the changes made by the search algorithm so the user can evaluate and compare how the plan changed after each search (Goals 4 &10). The Plan Cycle Option allows the user to watch all of the moves made by the search algorithm (Goals 4-7). The four remaining tools, the Pie Chart Control, Sensitivity Analysis Display, Metric History, and Weight History were developed in a Microsoft Excel 2003 spreadsheet with macros programmed in Microsoft Visual Basic 6.3. The Pie Chart Control replaces the Sliding Bars as a method for entering the weights. It accurately represents the relationships between the objective function weights (Goals 1-4). The Sensitivity Analysis Display shows the user how much a weight must be changed to guarantee a new solution (Goals 4 & 11). The Metric History chronologically graphs the metrics of all the plans discovered by the user (Goals 4 & 8). The Weight History shows the weights used to steer the algorithm. When used together, the Weight and Metric Histories reveal the cause and effect relationship between input coefficients and the plans discovered by the user (Goals 2,3,4,9, & 10). The Excel spreadsheet was used as a prototype to design and implement these tools quickly to test them experimentally before full integration with the HGA. The spreadsheet and HGA interface require separate computer windows to be visible to the user at all times.

Each of the six tools accomplishes at least one of the above goals and all of the tools are designed to meet Goal 4, that is, to improve the user's mental model. The combined effect of the tools is intended to accomplish all of the above goals of TBD for controls and display for this specific HGA. The tools improve user understanding and ability to interact with the algorithm;

however, it was not possible to incorporate trust information. Information on how to trust the models, metrics, and applicability to the real world were not addressed with these tools because the HGA was a test platform to evaluate human-machine collaboration, and not meant to be used in real-world UAV routing.

4.5.1 Pie Chart Control

A Pie Chart Control was designed to replace the Sliding Bars to better reveal to the user the relationship between the objective function coefficients. As seen in Figure 13, the Pie Chart Control is labeled as the "Objective Function Guidance Tool" on the spreadsheet and is more than just the pie chart graphic. The panel on the left side of the spreadsheet allows the user to type in weights. The pie chart graphic on the right side of the spreadsheet displays the relative proportions of the weights that steer the search algorithm, which are not displayed when using the Sliding Bars. The Pie Chart Control accomplishes TBD Goals 1-4 by improving HGA controllability, increasing objective function transparency, and providing helpful information to enrich the user's mental model of the HGA. It allows users to steer the HGA by thinking about the numerical weight values or visualizing them graphically on a pie chart.

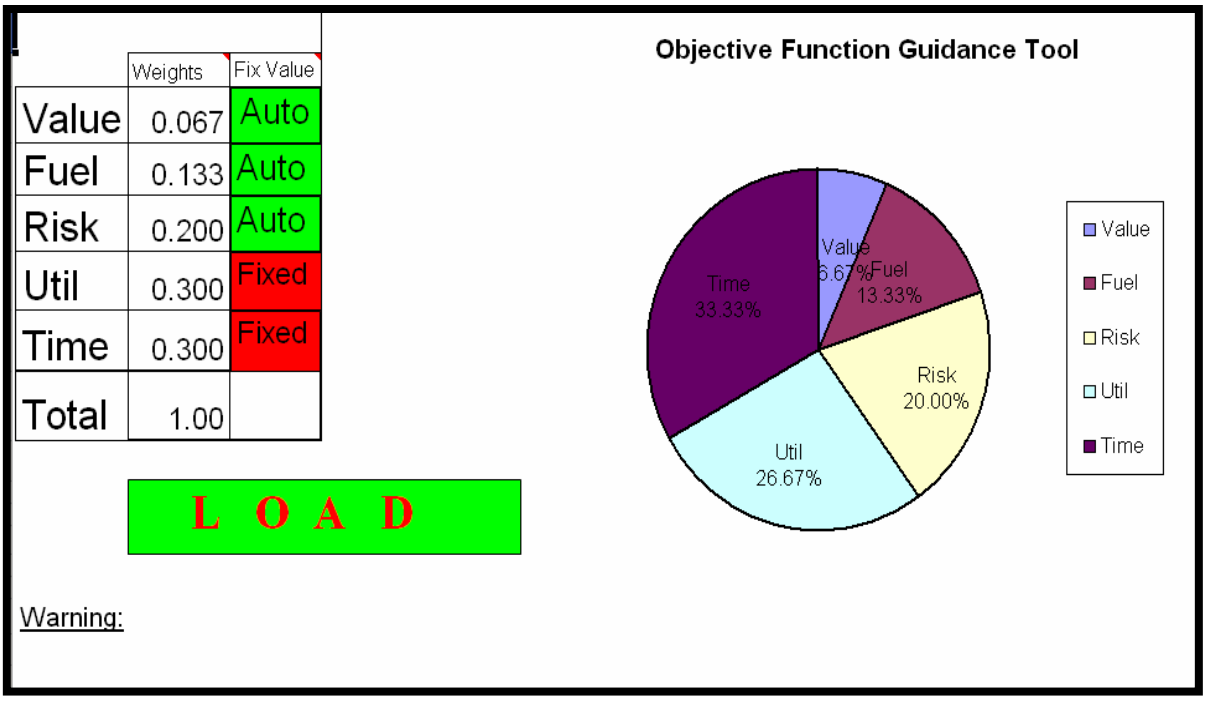


Figure 13: Pie Chart Control – Level 1
In addition to the pie chart graphic, the Pie Chart Control includes everything shown above. When the input weights do not sum to 1, the 'Load' button is replaced by the 'Adjust' button.

The user can enter any positive number, including zero, which allows him/her to think of the weights in various ways. For example, a user who thinks fuel is three times as important as value, time is half as important as value, value is equal to risk, and utilization is unimportant can set the weights: value = 1, fuel = 3, risk = 1, util. = 0, time = .5. A user can also specify the weights on scales from 0 to 100, like the Sliding Bars, or as percentages so they all sum to 1 or 100. This allows someone to think about and enter the weights in a method convenient to his/her own thought process. When the 'Adjust' button is selected, the interface shows how the algorithm interprets the weights, by scaling them so that they sum to 1. This provides a consistent means of conceptualizing the ratios of the weights and displays them on the pie chart graphic. 'Adjusting' the weights refers to the user selecting the Adjust button, while 'changing' the weights refers to a user changing their values and therefore their relative proportions. A user can continue to change and adjust the weights until satisfied. Finally, the user clicks on the 'Load' button to set the adjusted weights as the coefficients for the next search. He can then begin the search on the HGA Interface by selecting the Local Search button. A pop-up window on the Excel spreadsheet reminds the user to start the search after the weights are loaded.

Next to each weight is a 'Fix Value' button that allows the user to 'fix' the weight by holding it constant when adjusting the other weights. To use this feature, the fixed value must be between 0 and 1, the sum of the fixed values must be less than or equal to 1, and no more than four weights can be fixed. Weights that are not fixed are labeled as 'Auto', meaning that they are automatically scaled so that all weights sum to 1 when the Adjust button is clicked. When using the Sliding Bars, a change in one weight is equivalent to changing all of the weights. The Fix Value capability allows the user to change some weights while keeping other weights constant. The Pie Chart Control provides more controllability than the Sliding Bars by allowing fixed weights to remain constant while adjusting 'auto' weights. If at any time the user enters negative values or sets fixed values that violate the above rules, a warning is displayed on the screen to notify the user of the mistake. Figure 14 below shows an example of this capability.

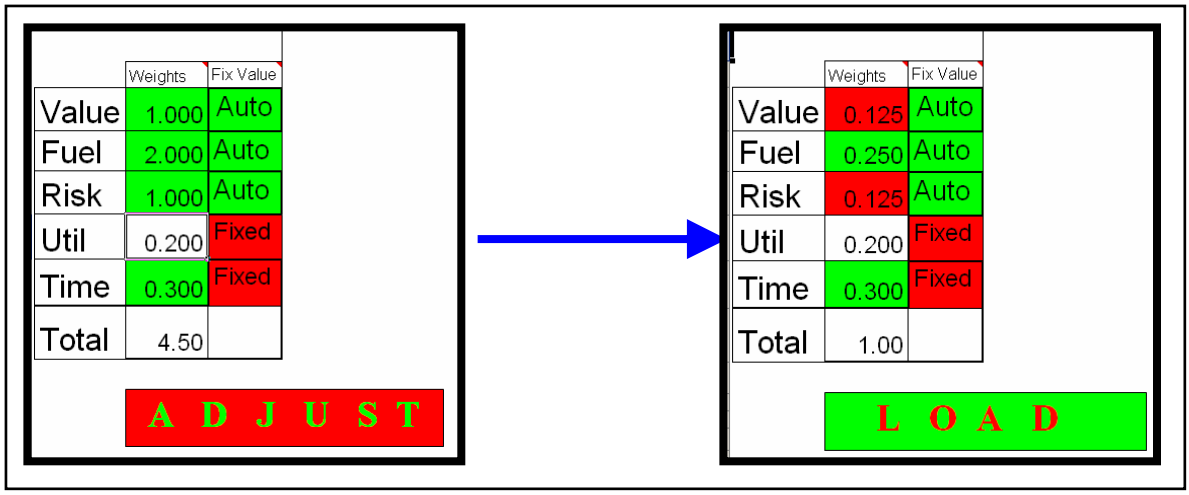


Figure 14: Example of 'Fixing' Weights
The user fixes utilization and time. When the Adjust button is selected, the value, fuel, and risk are scaled so that their ratios remain the same and all of the weights sum to 1. Utilization and time remain at their current values. Now that all the weights sum to 1, the Load button replaces the Adjust button.

The background color of the weights show if they have increased, decreased, or remained the same from the previous search. A green background indicates that the weight is larger than the previous search. The red background indicates the weight is smaller. The white background indicates that the weight is unchanged from the previous search. The background colors are only accurate after the weights have been adjusted. Background color provides a reasonable prediction of how the next plan will differ from the current plan. For example, a green background predicts an improvement of that metric, wile a red background predicts a deterioration of that metric.

This Excel prototype did not include the ability to use the mouse to change the weights by clicking on the pie chart graphic and changing the size of the pie slices. Directly manipulating the pie chart graphic removes the necessity of adjusting the weights. Because percentages are displayed on a pie chart, no matter how the weights are scaled, the pie chart graphic looks the same. These percentages, in decimal form, are identical to the values calculated when the weights are adjusted. The use of a more advanced GUI designer kit would allow the user to change the weights with the pie chart graphic and maintain the ability to fix pie slices to constant sizes. It would also make it possible to improve the interfacing between the prototype tools and the HGA GUI. As an example, the user could begin the search by selecting the Load button (renamed the “Search” button), rather than selecting Load and then clicking the Local Search button on the HGA GUI.

4.5.2 Sensitivity Analysis Display

The Sensitivity Analysis Display, shown in Figure 15, adds information to the Pie Chart Control. The purpose is to expose the magnitude of a change in the coefficients required for a change in the resulting solution during the operator input stage. A minimum and maximum value required to produce a change in the solution is calculated for each weight. On the GUI, this range is labeled the "No Change Range" (from here forward this will be referred to as the sensitivity range). Similar to the objective function weights, the sensitivity analysis is displayed numerically and graphically. Three assumptions must be met for the sensitivity analysis to be accurate:

1. Only one weight is changed.
2. None of the other weights is fixed.
3. The weight only needs to be outside the sensitivity range *before* adjusting the weights. A fixed weight causes a greater weight change than is necessary. For the smallest necessary weight change to create a new plan, 'auto' should be selected.

If the above assumptions are violated, a new plan might still be generated, but the exact values for the sensitivity range cannot be guaranteed. However, it is guaranteed that if only one weight is changed, and it is still within the sensitivity range, a new plan will not be generated in the next search.

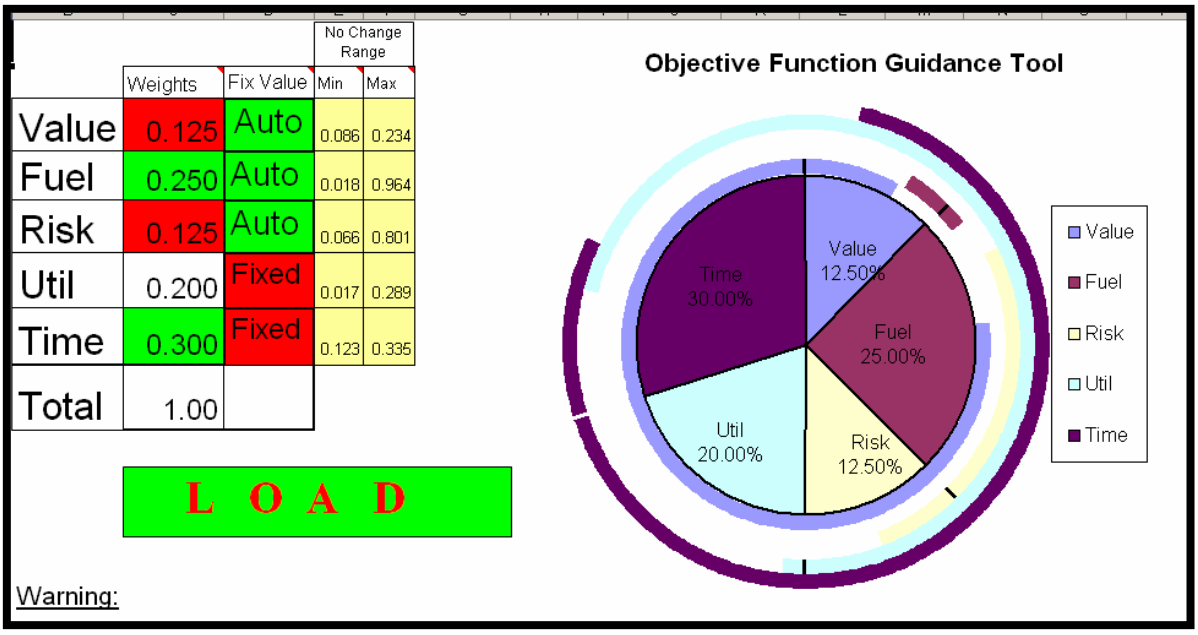
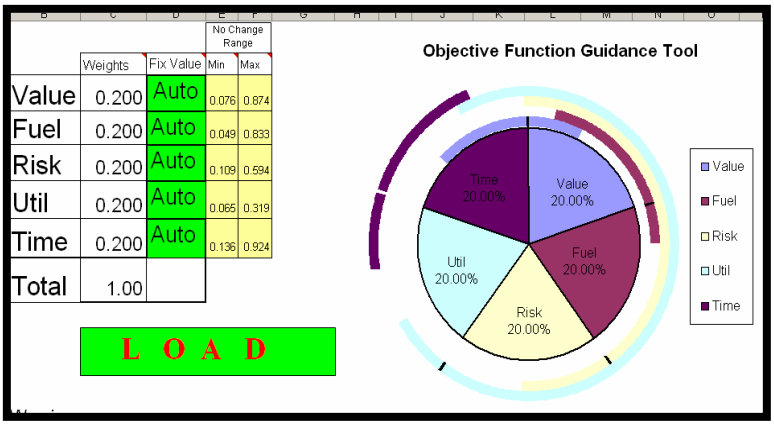
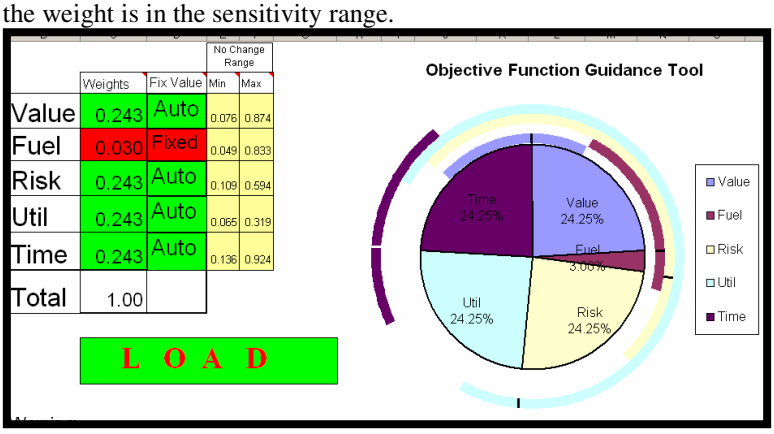


Figure 15: Pie Chart Control with Sensitivity Analysis Display - Level 2

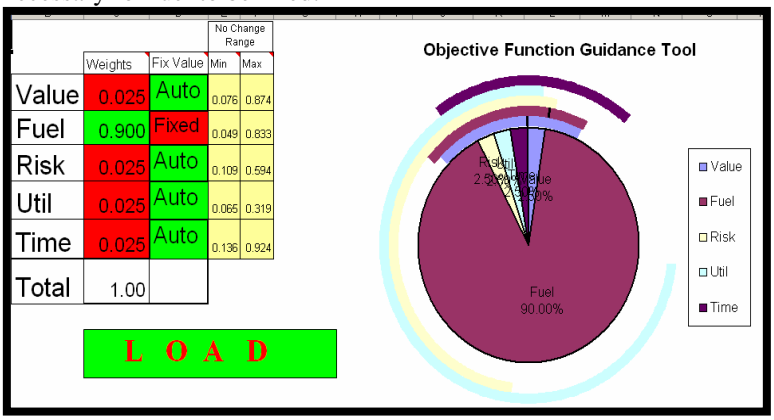
The numerical values of the sensitivity range are displayed next to the Fix Value buttons. These values are also represented graphically with colored rings around the pie chart graphic. The purpose of this graphical representation is to allow the user to adjust a pie slice with the mouse outside the sensitivity range, without having to refer to the numerical values. Each ring of sensitivity analysis is color coordinated to match its respective pie slice. A tick mark appears on each ring that corresponds to one of the edges on its respective pie slice. This edge of the pie slice is considered stationary. The other edge of the pie slice can then be used to adjust the size of the slice, and thus change the weight. If the adjustable edge is moved within its respective colored ring, the pie slice exits the sensitivity range. The same three sensitivity analysis assumptions apply to the manual manipulation of the Pie Chart Control. See Figure 16 below for an illustration:



Panel 1: After a search, all weights are outside their respective sensitivity rings. This can be seen by looking at the adjustable edge of the pie slice (the one without the tick mark). The location of the adjustable edge denotes whether the weight is in the sensitivity range.



Panel 2: Fuel is set inside its sensitivity ring, less than the minimum value, before being adjusted. Note: it is not necessary for fuel to be fixed.



Panel 3: Fuel is set above maximum value – inside the sensitivity ring.

Figure 16: Example of Interpreting Graphical Sensitivity Analysis – Level 2
Only the fuel weight is changed in Panel 2 and 3 to illustrate the sensitivity analysis graphic. To guarantee a new plan the movable edge of one pie chart slice must fall within its respective sensitivity analysis ring, *before* adjusting. The three assumptions must be met to guarantee a new plan.

Because the current pie chart slices cannot be changed with the mouse, the visual display of the sensitivity analysis might not be very useful for the current prototype.

The purpose of the Sensitivity Analysis Display is to prevent the user from making small changes to the coefficients that do not generate plans different from the current plan. This accomplishes TBD Goal 11, but comes at a cost. It can take between 10-30 seconds, or longer, to generate the sensitivity range.

To calculate the sensitivities for each weight, two binary searches are performed, for a total of ten binary searches for each set of weights. Before the binary searches begin, an initial upper bound is found by multiplying the current weight by 2 until a plan change is guaranteed or the weight reaches a maximum value of 100. This guarantees that the upper bound in the binary search is large enough to change the plan, unless it reaches the maximum value. This preprocessing adds five additional searches to the process. To find the maximum value in the sensitivity range, the binary search uses the computed upper bound and the current weight as the lower bound. The search converges to a set tolerance of $1.0 * 10^{-3}$. To find the minimum value in the sensitivity range, a binary search sets the lower bound to 0 and the upper bound to the current weight. It also converges to the same set tolerance. Every iteration of the ten binary and five preprocessing searches requires that the upper bound (preprocessing search) or the midpoint of the lower and upper bound (binary search) be tested to see if it changes the current plan. Each of these tests requires a neighborhood of plans to be generated. The neighborhood must then be searched to see if at least one improving move exists. This time-consuming process occurs tens or hundreds of times for each set of weights. The HGA interface displays a "Calculating Sensitivities" notice next to the Local Search button to inform the user when the sensitivity analysis is being calculated.

While the Sensitivity Analysis Display provides helpful information to the user, the display is advantageous only if its information is worth its computational time.

4.5.3 Rationale Window Display

A Rationale Window, shown in Figure 17, was designed to explain why the current plan was selected by the HGA as the best solution and how it outperformed the previous solution. The Rationale Window is labeled the "Algorithm Rationale for Plan Change" and is located on the

right side of the HGA interface. The purpose of this display is to achieve TBD Goal 10 by linking the user's weight inputs to their effect on the search process.

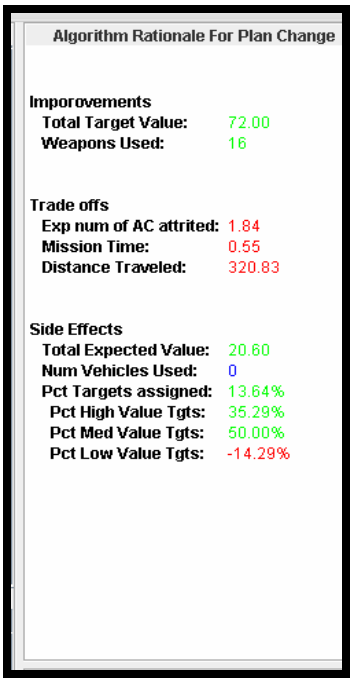


Figure 17: Rationale Window – Level 1

From the perspective of the search algorithm, every new plan is an improvement from the previous plan. Even if the user is not satisfied with the new plan, or thinks it is not an improvement, the search uses the user's new weights to find improving moves that increase the plan's score. The five metrics used to calculate the plan score can be broken into two categories: improvements and trade-offs. The improvements are metrics that have changed to increase the score from the previous plan. Increases in total value and utilization, and decreases in time, expected attrition, and distance, yield improvements. Since the plan is considered an improvement over the previous plan, trade-offs are metrics that have decreased the plan's score from the previous plan, but are justified by the improvement metrics. A third category of metrics is side effects. These metrics are not considered in the objective function or plan score, but might be important to the user and can be influenced by changing the weights.

These three categories are displayed in the Rationale Window, and the metrics are displayed in the appropriate category. Metrics that have a positive effect on the plan are colored

green, negative effect - red, and neutral effect - blue. The values displayed only show their change from the previous plan, with the units being the same as those in the plan metric window.

The Rationale Window shows how much the plan has changed and what considerations, in terms of improvements and trade-offs, the algorithm has made. It also distinguishes metrics that the algorithm does not consider and that the user cannot directly influence – the side effects. The Rationale Window also links the weight changes to the metric changes in the plan. Weights that have increased and are highlighted in green on the Pie Chart Control are expected to improve their respective metrics in the next search, and be highlighted and placed under the improvements category in the Rationale Window. Similarly, weights that have decreased and are highlighted in red on the Pie Chart Control will likely cause their respective metrics to be highlighted in red as trade-offs on the Rationale Window. The user can also observe how the weights have changed the metrics in the side effects category. Due to the nature of the local search, the plan changes might not always align with predictions provided by the Pie Chart Control; however, drawing attention to how the user's input influences algorithm output is necessary for the user to build appropriate trust in the HGA.

4.5.4 Plan Cycle Option Display

The Plan Cycle Option, shown in Figure 18, allows the user to watch graphically the improving moves made during the search. One drawback is that the frequent redrawing of the plan graphic screen slightly slows the algorithm. The Plan Cycle Option is labeled as the "GUI Refresh Delay" and is located on the bottom right side of the HGA interface, underneath the Rationale Window. The user can turn this feature on or off. When on, the user can select the length of the GUI refresh delay (0-20 seconds). The Rationale Window updates with each new plan displayed to show how the current plan has changed from the original plan that began the search. The purpose of the Plan Cycle Option is to show the user the exact moves made by the algorithm. The user can change the time of the refresh delay depending on how long he would like to study each intermediate plan. The Plan Cycle Option accomplishes TBD Goals 4 – 7 by providing additional information about the algorithm's progress and process and giving the user control of when he wants this information.

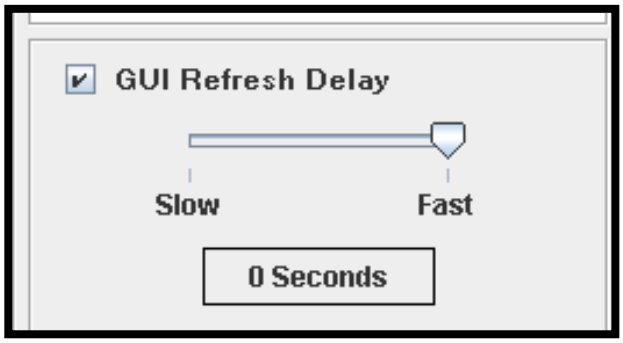


Figure 18: Plan Cycle Option – Level 1

4.5.5 Metric History Display

The Metric History tracks and displays five plan metrics during the entire human-guided search. It is labeled as the "Scaled Metric History" and is located on the bottom right corner of the Excel spreadsheet. In this prototype, value, attrition, distance, time, and expected value are tracked because these are usually the most important metrics to the user. However, allowing the user to select which metrics to track would provide the user with more metric options and the ability to track only necessary metrics. The purpose of the Metric History is to accomplish TBD Goal 8 and show the user the known solution space. This information can be used to compare previous plans against the current plan and keep track of the HGA performance.

Each metric is standardized to a value between 0 and 1 and then plotted on the graph. For each metric, the largest known value, out of all the plans displayed on the Metric History, is scaled to 1 and the remaining values are proportionally scaled between 0 and 1. The x-axis displays the number of searches performed, and the y-axis shows the standardized values of the metrics. Colored lines connect the metrics to track how they have changed over the various searches. The metrics are continually rescaled to reflect new information about the solution space and to allow the plans to be compared against one another. A key on the right side of the graph shows what color refers to what metric. An arrow points up or down, showing what direction the metric must change to improve the plan. The arrows for value and expected value point up while the attrition, distance, and time arrow point down.

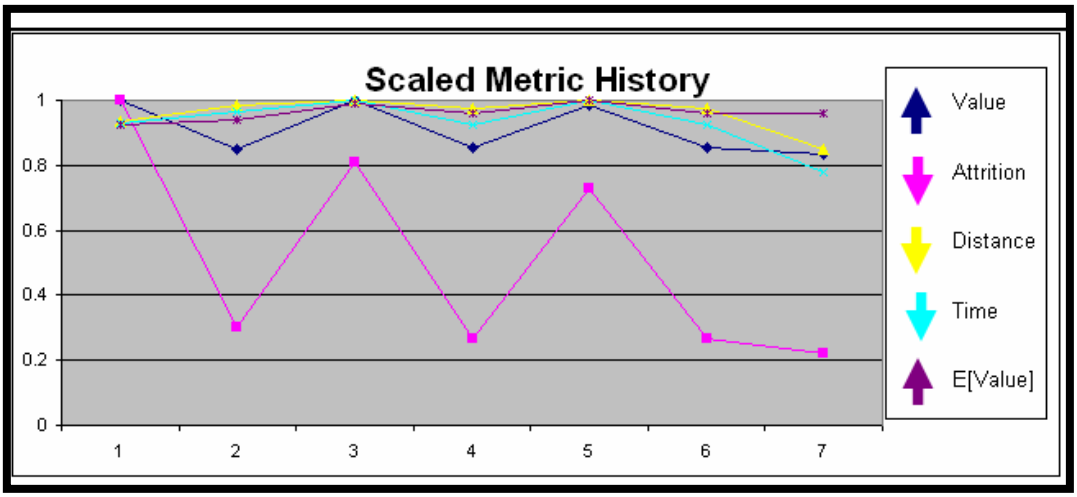


Figure 19: Metric History – Level 2

Using Figure 19 as an example, the following information can be learned from the graph. Over the seven searches, expected attrition was reduced by nearly 80%. The expected value from Plan 7 is only 5% less than its highest value in Plan 5. Distance and time in Plan 7 are at their lowest value over the seven searches and were reduced by about 15% and 20% respectively. It is also possible to conjecture how the changes in one metric are related to the changes in another metric, however, the weight changes must also be considered to draw any definite conclusions.

4.5.6 Weight History Display

The Weight History tracks and displays the weights during the human-guided search in a manner identical to the Metric History. It is located on the bottom left corner of the Excel spreadsheet. The Weight History is intended to be used in conjunction with the Metric History. By comparing the information on both history graphs, the user can analyze how his control inputs have resulted in different plans. This information reinforces that provided by the Pie Chart Control and Rationale Window, but for the entire search. The user can also use this information to predict how future changes in the coefficients might have an effect on the solution. The combined effect of comparing these charts accomplishes Goals 2, 3, 4, 9, and 10.

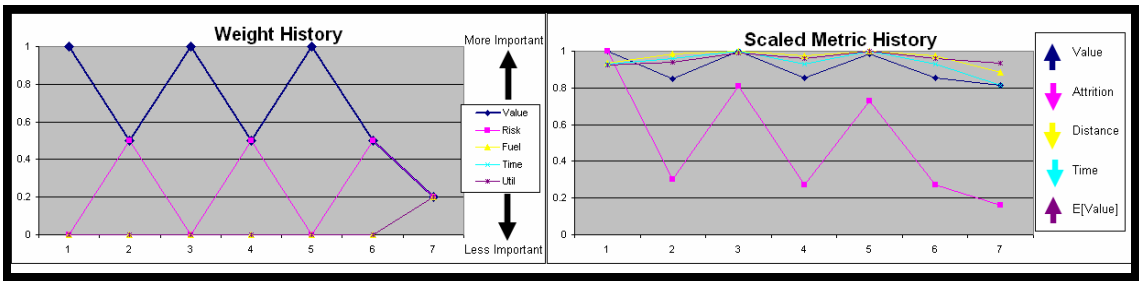


Figure 20: Side-by-side View of Example Weight and Metric Histories - Level 2

Referring to Figure 20, the search strategy of alternately setting value to 1 and then value and risk to .5 reduced the expected attrition metric in searches 1 - 6. The strategy stopped being effective after the 5th search, because Plan 6 is identical to Plan 4. Plans 3 and 5 both have the highest expected value, but plan 5 has a lower expected attrition. Search 7 sets all the weights to .20, which decreased all of the metrics to their lowest value in the search. Notice how the user improved the solution by Plan 7. Plan 7 had a slightly higher expected value than Plan 1 and a lower expected attrition, distance, and time. Value is 10% lower, but for the purpose of this scenario, expected value is assumed to be a more reliable metric. The combination of these two graphs might lead to the development of human-guided search strategies and heuristic methods that direct the search to desired areas of the solution space, escape local maximums, and improve the quality of the HGA solutions. It might take users considerable time and experience with the software and the specific vehicle routing problem to develop efficient search strategies.

Despite the simple design of the Weight and Metric Histories, it might be difficult to interpret and compare the data on the two charts. It is possible that Draper Laboratory's Decision Space Visualization (DSV) tool, discussed in Chapter 2, can improve the display of this information. The DSV provides the user with a greater capability to interact with the display and might accomplish the TBD goals better than the Weight and Metric Histories.

4.6 Applying TBD of Trust Information

This section presents the application of the trust information and calculation approach, described in Chapter 3, to UAV Routing HGA.

4.6.1 Abstract Hierarchy

An abstract hierarchy was developed to reveal the relationship between the components of the HGA. A high-level diagram is shown below and a detailed explanation follows:

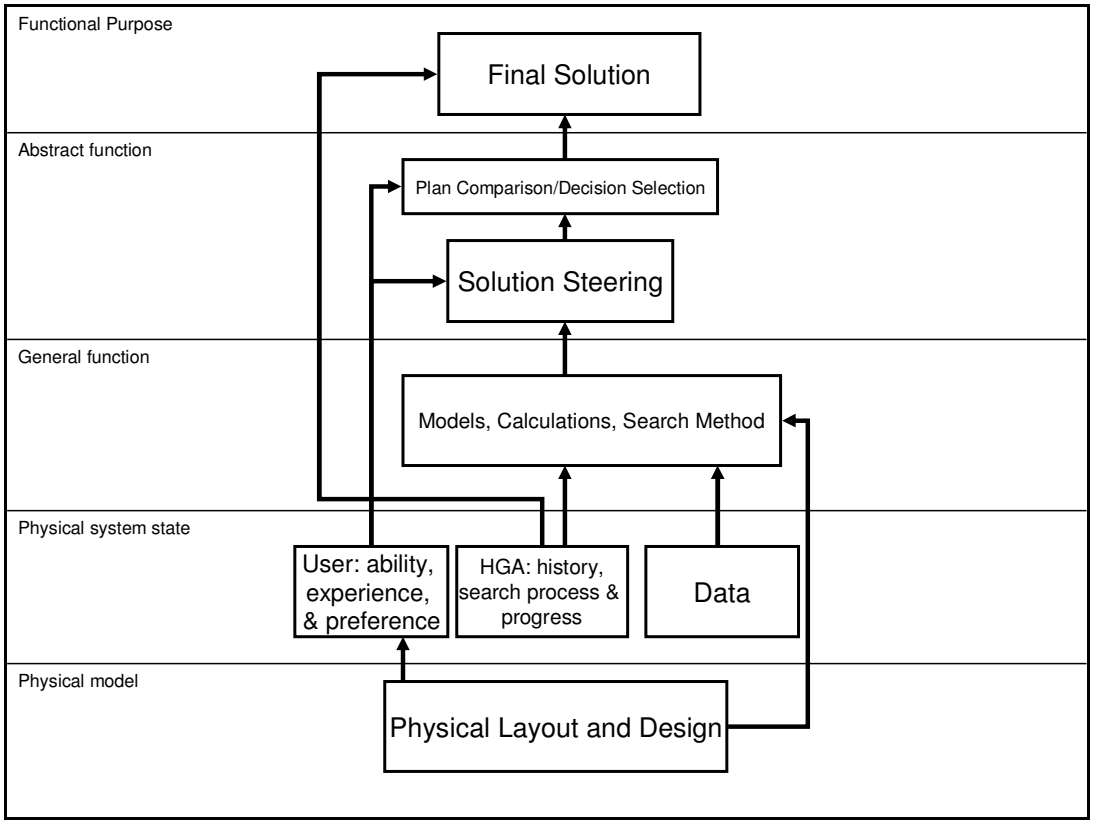


Figure 21: High-level Abstract Hierarchy for the UAV Routing HGA

Starting at the lowest level, the Physical Model's layout and design of the system includes four elements:

1. Computer software and hardware.
2. Communication between software applications such as the Excel spreadsheet and HGA interface.
3. Efficiency of the computer code to implement the algorithms.
4. Layout of controls and display for human-machine collaboration.

The first three elements influence the models, calculation, and search method in the General Function. The fourth element affects the user's ability to collaborate with the HGA, specifically in the solution steering and plan comparison/decision selection step in the Abstract Function level.

The Physical System State level contains three components: user, computer, and data. The user component includes the user's preference for specific types of solutions, experience with and knowledge of the HGA, qualification for decision making, and knowledge of the problem domain. The computer component can be broken into three elements: its current state, past progress, and historic performance. The first element includes the methods currently being performed by the computer and the current state of the solution. The second element is the algorithm's past search processes and progress made to improve the solution and past plans generated to solve the problem. The third element is the HGA's historic ability to solve similar problems. The user's knowledge of these three elements improves his knowledge of the HGA and therefore his ability to implement solution steering. The state of the system also affects the search methods currently being performed and how they can be used in the future. The third component, data, is the information that is directly fed into the models, calculation, and search methods. The quality of the data influences the reliability of the methods in the General Function.

The General Function level contains the models, calculations, and search methods in the HGA. These include the model for assessing UAV risk, the search and the moves it can make, calculations for all the metrics, distance calculation including the assumption of straight-line distance, etc. These elements have their own reliability and measure of applicability to real-world UAV operations. The outcome of these methods directly affects how the user steers the HGA in the Abstract Function level.

The Abstract Function includes the two main components of human-computer interaction: solution steering through changing the weights, and comparing and selecting a final plan. The quality of the calculated metrics used in the HGA's objective function affect how well the user can steer the algorithm. For example, consider a user that requires the UAVs to travel short distances. If the user heavily weighs the fuel coefficient to generate plans of low distance, but the straight-line distance calculation is a poor real-world assumption, the plans generated might poorly represent his objective. The user will generate plans of low distance and expect

them to perform as planned, but actual implementation might yield unexpected results. However, providing the user with trust information about the quality of the distance calculation allows him/her to steer the HGA to solutions that do not heavily weigh this highly variable metric. Trust data establishes an honest expectation for HGA quality, which will affect how a user steers the HGA to generate plans. The plans generated through solution steering directly influence the pool of plans from which the user selects the final solution. The plan comparison and decision selection component affects how well the user can compare these plans and confidently select the best one as the final solution in the Functional Purpose level.

Each element discussed above can be assessed and linked to other elements in the same component or in other levels of the hierarchy. As an example, some of the most highly variable metrics in the HGA include the expected attrition and expected value metrics. These metrics can be very helpful to the user, but as can be seen from the detailed hierarchy, they are subject to many variables. Starting from the top of the hierarchy, the expected attrition metric is a product of the expected attrition calculation. Expected attrition is calculated by summing the risk accumulated by UAVs traveling their routes. The expected value metric is calculated in a similar manner. Expected value is calculated from the accumulated risk and target value as the UAVs travel their routes. The magnitude of the risk and value metrics are directly affected by the human-guided objective function weights and the search. The time, utilization, and distance metrics that are also directly affected by the objective function are labeled as 'other metrics.' In addition to the risk and value weights, the objective function contains weights for utilization, time, and fuel that also influence the risk and value metrics. All of the metrics considered by the objective function are determined by their respective models.

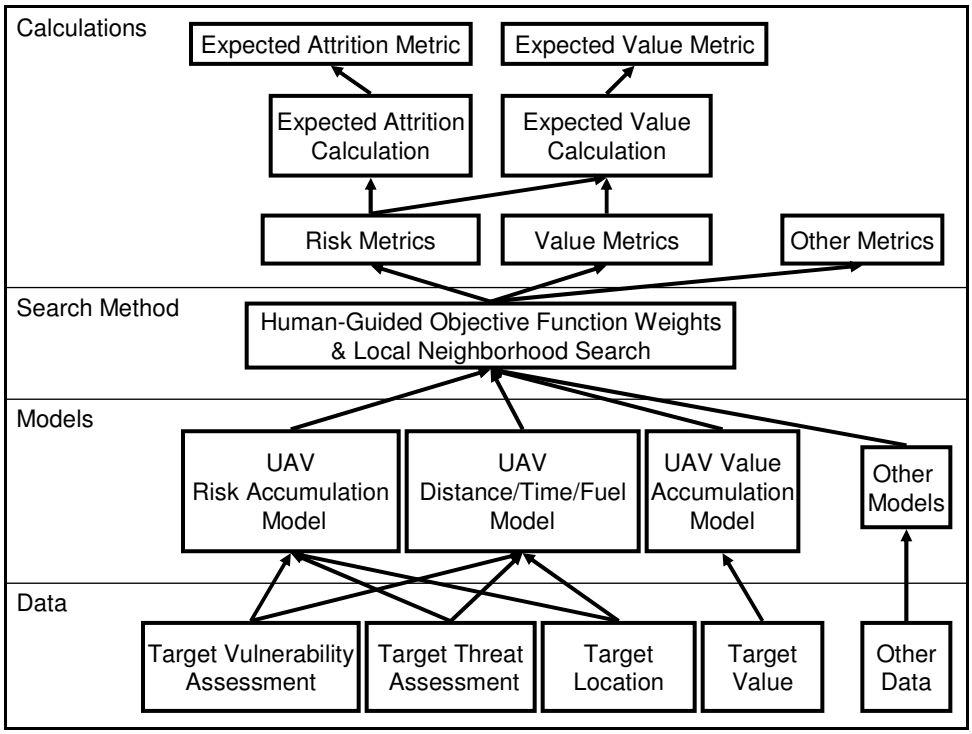


Figure 22: Detailed Breakdown of Expected Value and Expected Attrition Calculations

The risk accumulation model determines how UAVs accumulate risk during the mission. This includes the assumption that UAVs prosecuting a SAM accumulate an insignificant amount of risk from that SAM due to jamming capabilities. The distance, time, and fuel models are combined as one model in the diagram. They can be combined because UAV route distance is directly correlated to the time it takes to traverse the route and fuel is directly correlated to time because of the fuel constant burn rate assumption. The straight-line distance assumption and constant velocity assumption also play a role in the accuracy of the model. The UAV value model simply sums the target values as the UAV traverses its route. The 'other models' box is added to account for metrics in the objective function not discussed above, such as utilization.

Data fed into the models can affect their accuracy. Three types of information are crucial to the accuracy of the risk accumulation and distance/time/fuel models: target vulnerability assessment, target threat assessment, and target locations. Target vulnerability determines how many munitions are required to destroy a target. The threat assessment determines the size and intensity of threat radii. The target location is necessary for the UAVs to locate their targets. If any of the above information is incorrect, it may affect the accuracy of the final plan. Some UAVs are dependent on other UAVs to destroy threatening targets so that their own vulnerability

is reduced. If one UAV's timing is off, problems can propagate though the entire mission and have a drastic effect on its outcome.

4.6.2 Display of Trust Information

As shown by the detailed hierarchy, the expected attrition and value metrics are dependent on many variables such as specific data, models, human inputs, search methods, and calculations. A full detailed hierarchy of the entire HGA is too much information to present to the HGA operator. Only relevant information should be presented to the user at his request or when the computer deems it necessary. In this case, displaying high-level connections between unreliable data and models used to make calculations might be sufficient. This will allow the user to be appropriately wary of highly variable calculations. Determining the correct level of trust information to display is necessary to prevent the user from becoming overwhelmed. The user does not need to understand the intricate details of the HGA. The user's expertise should be in UAV operations and mission planning, not Operations Research methods for a UAV routing problem. The design of the HGA must allow the user to employ his expertise to generate high quality and appropriately trusted solutions. Supplying the user with too much information could hinder the utility of the HGA.

A possible method of supplying trust information is with an adaptive display. The display notifies the user of additional HGA complexities and provides appropriate trust information only when necessary. A drawback of an adaptive display is the additional level of automation it adds to the system.

Models in the HGA may need to be updated or changed depending on the data or problem situation. The current distance model might be appropriate for high altitude UAV operations, but infeasible for operations in low altitude mountainous terrain. In mountainous terrain, Thomas Krenzke's (2006) ant colony genetic UAV routing model can be implemented. A future version of the HGA may include allowing the operator to select different models. Because different models have their own strengths, weaknesses, and differing trust implications, it is necessary for trust information to be provided that reveals how the final solution will be affected. In fact, models may be selected because of their known trust attributes in the specific scenario.

The detailed hierarchy shows the connections between various components, but the magnitudes of these interactions are unknown. For example, it is helpful to know how variability in target locations affects the calculated risk metrics. Assigning magnitudes to trust information for all HGA components can be accomplished in various ways, but the effort required to attain the information should be weighed against its utility. Historical information about the accuracy of the final solution and specific metrics can be tracked over time. Simulations can show how variability in the individual HGA components (models, calculations, data, etc) affects the accuracy of the final solution. Simulations may also determine how the variables interact, revealing how calculations can be performed to quantify the variable interactions. The computer can track the time spent by the operator to generate solutions and the size of the solution space searched. Supervisors, the computer, and user can collaboratively evaluate the operator's performance, qualifications to operate the HGA, and ability with certain components in specific circumstances. Because all of this information can affect the final solution, some means of measurement helps the user evaluate appropriate trust in the HGA and final solution.

An interesting option is for the computer to monitor the user's preferences for certain plan attributes. For example, the computer could warn a user, who selects plans based on their expected value, that the calculation has high variability due to data variability. The computer might also be able to make suggestions on how the user can make robust decisions to compensate for variability and uncertainty.

The display of the trust information depends on what information is available to display. An adaptable display, discussed earlier, displays portions of a detailed abstract hierarchy that the user should consider. A colored circle can be used to indicate the reliability of the metric calculations:

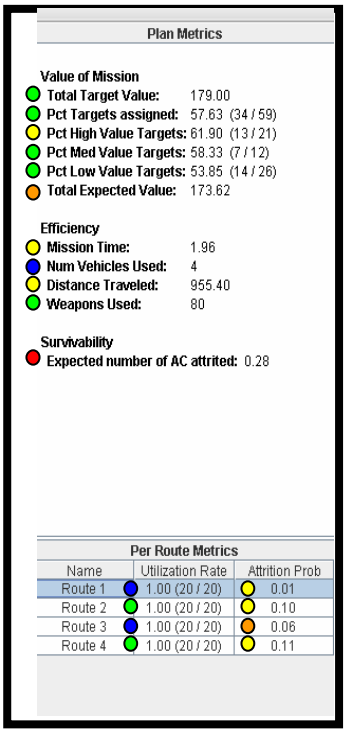


Figure 23: Display of Trust Information on Rationale Window
Colored circles show the user how much to trust the displayed metrics.

The operator could then select a colored circle for more detailed information, which may include the rationale for the reliability rating, a detailed hierarchy, and suggestions for interpreting and using the information.

HGA operators should be provided trust information during training. The users should be trained on how the HGA works, including viewing some of the detailed relationships displayed on an abstract hierarchy. The purpose of this training is to provide the user with an adequate mental model. During actual implementation of the HGA, trust information is provided to update the user's mental model based on the circumstances of the scenario. The user should not see any trust information about the HGA that they have not already seen in training. The trust information should ensure the user has an adequate mental model and enough information to evaluate the HGA and final solution. This trust information should be used to reinforce training and account for unique problem circumstances.

Before conducting further research into the feasibility of the trust information approach, it must be determined if trust information is helpful to the HGA operator. Providing trust information that users will not consider in their decision making, will not help them build appropriate trust.

Chapter 5

Experimental Evaluation of HGA TBD Components

An experiment was conducted to evaluate the trust-based design (TBD) tools. Participants were given four UAV routing scenarios that identified specific mission objectives. They then used the various tools to steer the UAV routing human-guided algorithm (HGA) to solutions that met their objectives. While the participants were searching the solution space, a global search algorithm searched the solution space using weights input by the user that best met his/her objectives. The participants then decided between implementing their human-guided solution, or the global search solution. After each scenario the participants evaluated the TBD tools' effectiveness in improving their understanding and trust in the HGA.

5.1 Participants

Eight participants served as the subjects for the experimental evaluation of the HGA and the TBD tools. All participants were Air Force 2nd Lieutenants who received their Bachelor of Science degrees from the United States Air Force Academy (USAFA). At the time of the experiment, they were Draper Laboratory Fellows pursuing Master Degrees from the Massachusetts Institute of Technology (MIT). Their training at USAFA included principles of warfare and the application of airpower. Their research at MIT focused on Operations Research, Aeronautical and Aerospace Engineering, or Engineering Systems. None of the participants had detailed knowledge of HGAs or previous experience with Human Machine Collaborative Decision Making (HMCDM) experiments.

5.2 Purpose

The purpose of the experiment was threefold. The primary purpose was to test the ideas of TBD for HGAs and to gain insight into the future design and use of HGAs in decision making. The

secondary purpose was to confirm the utility of HGAs as a viable decision-making tool. The tertiary purpose was to evaluate the need for HGA users to be directly provided with trust information. Trust information includes the reliability of the data, algorithm models, metric calculations, final solution, and data related to the HGA's historic ability to solve real world problems. Since the purpose of TBD is to improve trust calibration by fostering appropriate trust in HGAs, various components of user trust were evaluated. The six TBD tools were evaluated for their utility and effect on plan quality and the decision-making process.

5.3 Experimental Design

Three levels of TBD were used as independent variables. Level 0, the control, used the initial HGA interface, which did not include the TBD components. Level 1 added the Rationale Window display and the Plan Cycle Option to the Level 0 interface. It also replaced the Sliding Bars with the Pie Chart Control. Level 2 built upon Level 1 by adding the Sensitivity Analysis, Metric History, and Weight History. At all levels, the HGA and its related interfaces were referred to as the "HGA testbed."

The experiment required the use of two computer screens for Levels 1 and 2. On one screen was the HGA interface and on the other was the Excel spreadsheet. Shown below in Figure 24, Figure 25, and Figure 26 are screen captures of the different experimental levels. The initial HGA interface can be seen in Figure 10.

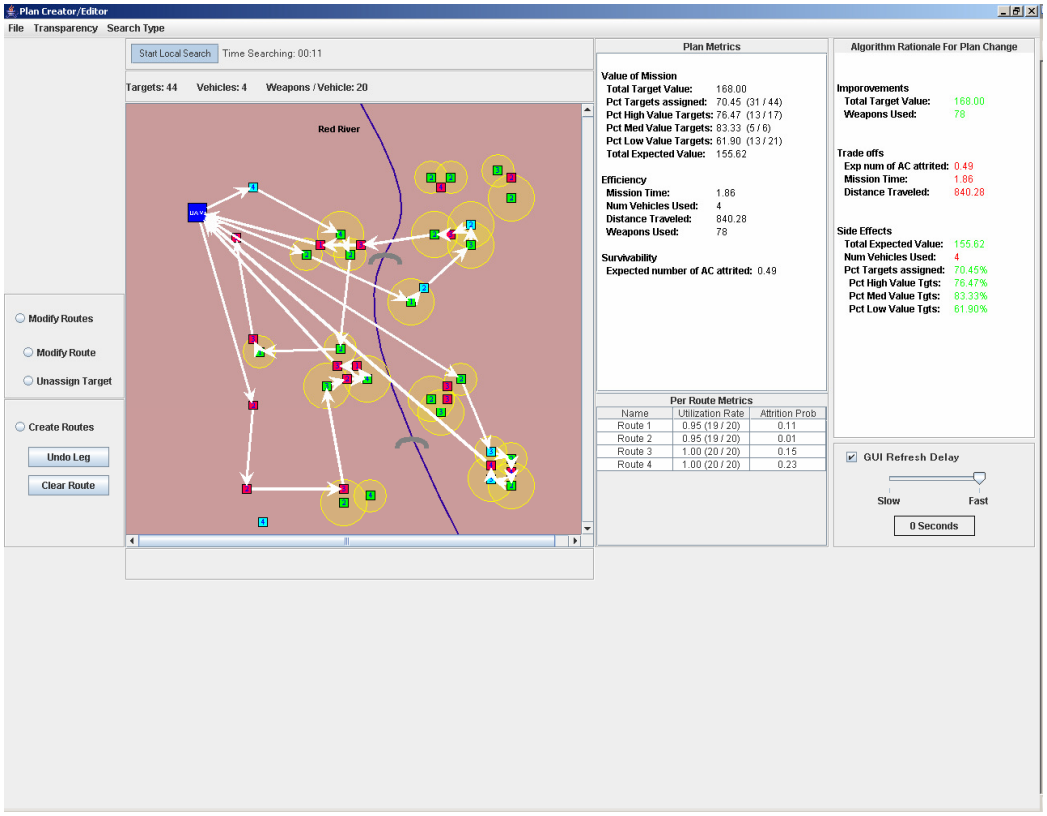


Figure 24: HGA Interface for Levels 1 and 2

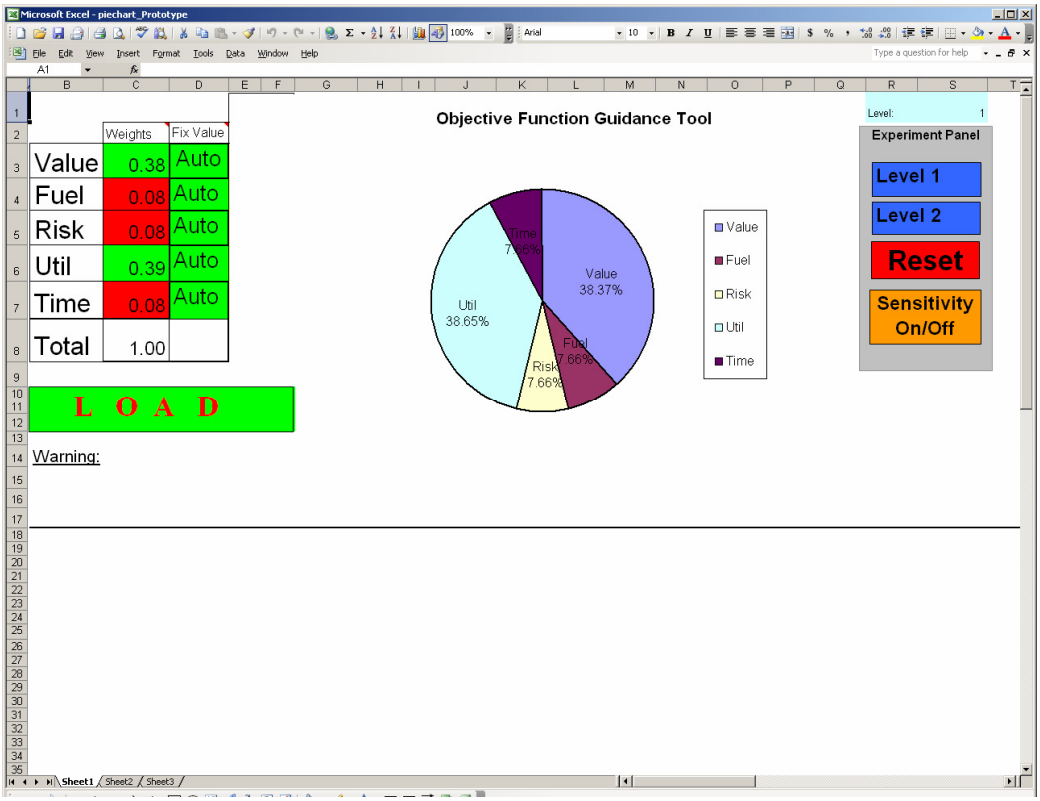


Figure 25: Excel Spreadsheet for Level 1

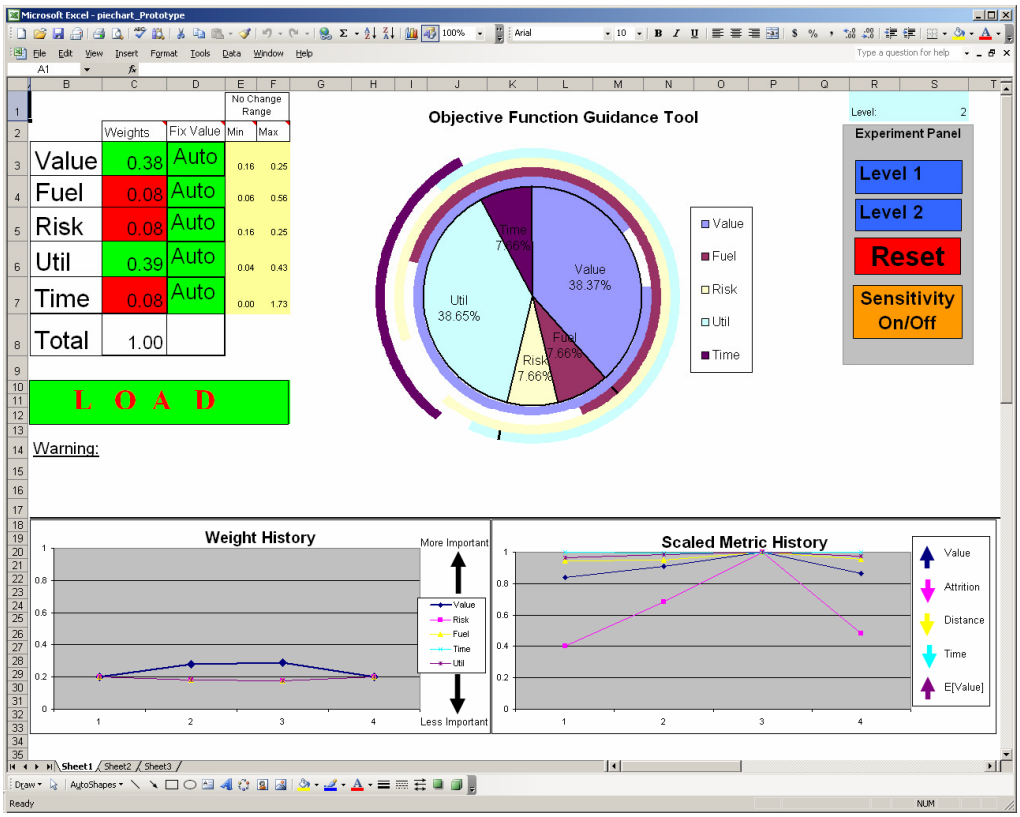


Figure 26: Excel spreadsheet for Level 2

The following is an outline of the experimental procedures used for each participant. Following the outline is a detailed explanation of the procedures.

1. a. Participant receives overview of experiment and Level 0 training
   b. Participant received training for randomly selected level of the first scenario
2. Participant reads scenario
3. Participant provides weights for global search
4. a. Participant begins 15-minute human-guided search
   b. 15-minute global search begins with the participant's coefficients
5. Participant finishes human-guided search and selects the HMCDM plan
6. Participant completes a questionnaire about the experimental level
7. Participant is provided the global and HMCDM plan and selects the better plan
8. Repeat steps 2-9 for the remaining experimental levels
9. Participant reads scenario 4

10. a. The six previous HMCDM and global plans are rotated and background graphics altered to match solutions to scenario 4
    b. Participant is presented with the six plans
11. Participant ranks the plans and writes an explanation of why he selected his number one ranked plan
12. Participant completes a post experiment questionnaire

Each experiment lasted between 2.5-3.0 hours. Most participants were provided with the Level 0 training guide the evening before the experiment to familiarize themselves with the experiment, UAV routing problem, and Level 0 interface. This significantly accelerated the participants' initial round of training required in step 1.

Four identical unmanned aerial vehicle (UAV) routing scenarios, shown in Figure 27, were modified from a previous experiment to be used in this experiment. A scenario consisted of a mission description and a map of the targets. Each scenario required the participant to assume the role of a UAV mission planner and route the four available UAVs. The mission descriptions were worded differently, their maps rotated, and background graphics altered so they appeared to be distinct. The scenarios emphasized accumulating as much value as possible, while ensuring the UAVs complete the mission in less than two hours and at most losing one UAV to hostile fire. The operators were told they could violate the two-hour constraint if they had good reason. Finally, the participants were informed that there was inaccuracy in the target locations and this could affect the reliability of the calculated plan metrics.

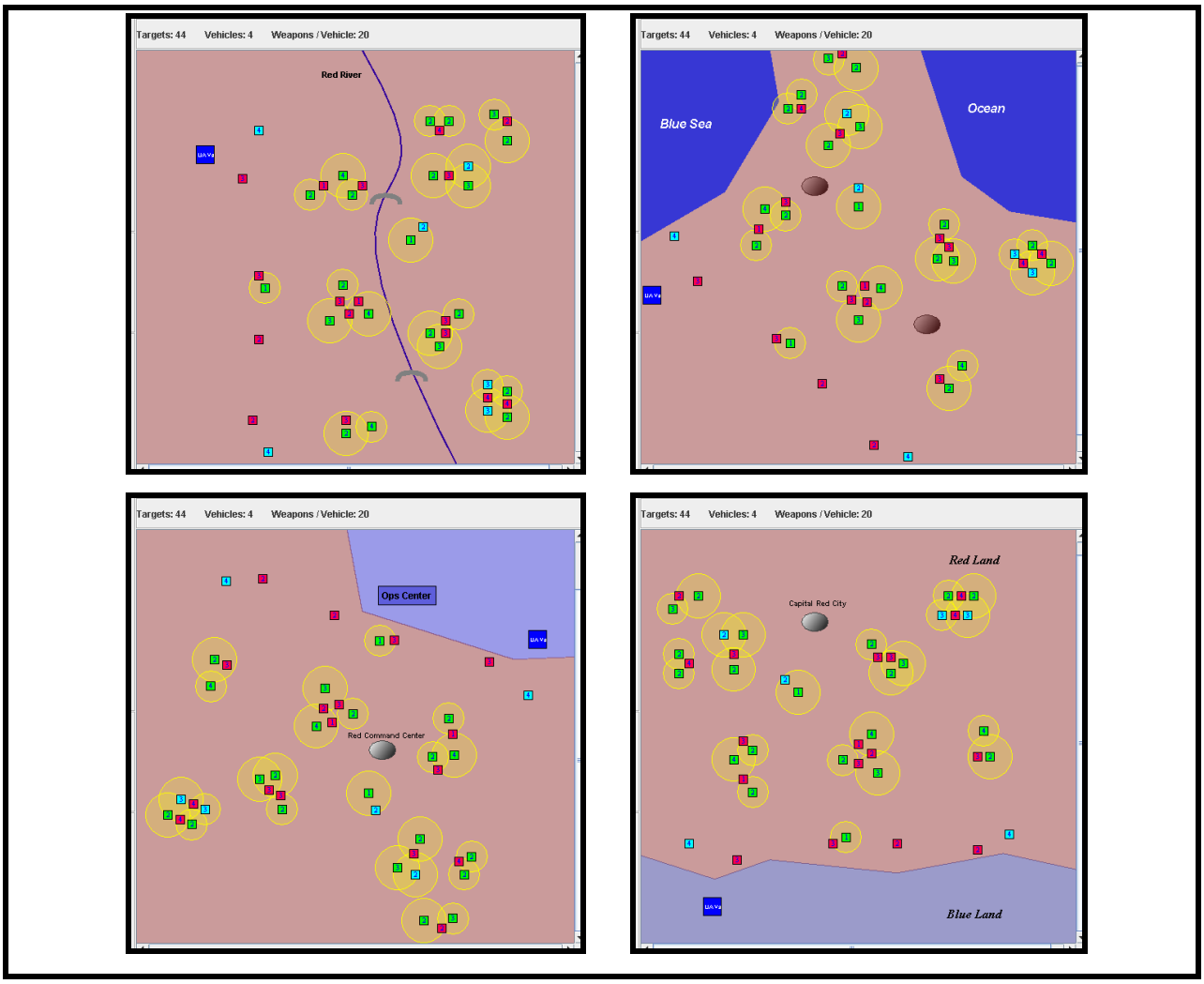


Figure 27: Scenarios Used in Experiment
All target locations are identical, but rotated and the background graphic changed to appear unique.

To test and compare the experimental levels, each participant completed the first three scenarios using a different experimental level. The order in which the levels were completed was random. For each level, the participants were given 15 minutes to search for high quality solutions to the scenario. During the human-guided search, a separate global search algorithm on another computer directed a local neighborhood search for 15 minutes. The participant selected a set of coefficients to steer the global search before they began their human-guided search.

To prevent the user from gaining an advantage from the clustering step used to generate the first plan in the human-guided search, the clustering step was removed. The initial starting plan for every HGA search was the smallest possible plan, the Null plan. Because a plan that

does not send any UAVs to destroy targets was not a viable solution, the solution that used the least resources (UAVs, time, attrition, fuel) was a plan that sent one UAV to prosecute the closest target to the UAV's home base. This plan is called the Null plan. By starting at the Null plan, the users had to steer the solution out of an area of poor plans and into an area that met their objectives.

The global search algorithm started at randomly generated seed plans. The local search made changes to these plans based on weights initially provided by the user. The user provided these weights, specifically for the global search algorithm, after reading each scenario. The algorithm kept track of the highest scoring plan and saved it as the 'Global plan' at the end of the 15-minute search. About 3000 plans were generated during each search, not including the tens of thousands of plans considered in the local neighborhoods of these plans. Due to the nature of this kind of search, there is no way to determine if the Global plan was actually the global optimum.

At the end of the 15-minute human-guided search, the participant had to select the plan that he thought would be the best solution to the scenario. This plan was labeled the 'HMCDM plan.' The participant had the option of using the Plan Comparison window for this step. The user was then provided with the global plan to decide whether he preferred the HMCDM or global plan. The decision between the HMCDM and global plan indicated which method of problem solving generated the best solution for the given scenario. After each level, the participant filled out a questionnaire.

The fourth scenario consisted of a comparison of the HMCDM and global plans generated in the three previous scenarios. These six plans were rotated to prevent the user from identifying them as the plans generated in previous scenarios. The user was presented with the plans in a Plan Comparison window and told to rank them from best to worst. This was another method to evaluate both the quality of the plans generated over the three levels of TBD and the utility of the HGA search.

The dependent variables included the effectiveness of the HGA, the usability of the TBD components, and user trust in the HGA. The effectiveness of the HGA was measured by the participants' selection of either the global or HMCDM plan after each scenario and the ranking of the six plans in scenario four. To compare the HGA testbed at the three levels of TBD, the questionnaire focused on the following:

- Search strategy
- HGA testbed's ability to support understanding of the algorithm's logic
- Predictability of how input weights changed the plan
- HGA testbed's ability to support the user in steering the algorithm to high quality plans
- Difficulty in operating the HGA testbed
- Dependability of the HGA testbed to find high quality plans
- Faith in the HGA testbed to cope with a large variety of scenarios
- Trust in the HGA testbed
- Confidence in the final plan

To evaluate the performance of the six TBD tools, the questionnaire asked about the following:

- Difficulty of use
- How it was used
- Effect on strategy
- Effect on understanding of the algorithm's logic
- How well it supported decision making

After the four scenarios, a final questionnaire probed the following topics:

- Trust in specific components of the HGA testbed
- Trust in data and plan metrics
- Using the mouse to manipulate the pie chart graphic
- Overall preference of the HGA testbed or global search algorithm
- Preference of TBD tools for HMCDM searches

The participants' plans were not compared against other participants' plans. When a participant assumed the role of UAV mission planner, they were given authority to use whatever qualitative

and quantitative metrics were available to evaluate plan quality. However, the participants were given objectives that emphasized the value/expected value, time, and expected attrition metrics as the primary criteria for evaluating plans. Because the participants had varying opinions on how to evaluate the plans, it was not appropriate to compare plan quality across the participants. The scenario 4 rankings allowed each participant to compare their own plans generated in the previous levels of the experiment.

5.4 Training

Initial training was provided to inform the participants of the problems they would be solving, how to use the HGA interface, and necessary details of the internal workings of the HGA. Additional training was provided before each experimental level on how to operate new tools on the HGA interface. The additional training did not provide any new information about the internal workings of the HGA.

The initial training included an explanation of the UAV routing problem, the metrics used to evaluate plans, and modeling assumptions made by the algorithm. The training explained how to enter weights and steer the HGA with the Sliding Bars. It emphasized how the weights are interpreted by the HGA with the example in Figure 28.

- You define the weights with sliding bars that range from 0 - 100.
- The values of the weights are not important, it is their relative proportions that the algorithm 'sees'.
  - For example, setting all the weights to 100 is identical to setting them all to 1.
  - The following four sets of weights are all identical:

	1	2	3	4
• Value:	5	10	75	100
• Fuel:	1	2	15	20
• Risk:	4	8	60	80
• Util:	1	2	15	20
• Time:	3	6	45	60

Figure 28: Explanation of How to Use the Sliding Bars

The training described the entire Level 0 interface and provided instructions on how to use it. An analogy of solution steering and an explanation of the experiment were provided before a

tutorial allowed the participants to practice using the HGA. The practice scenario is shown in Figure 29. The initial training provided the knowledge and practice necessary to properly operate the HGA.

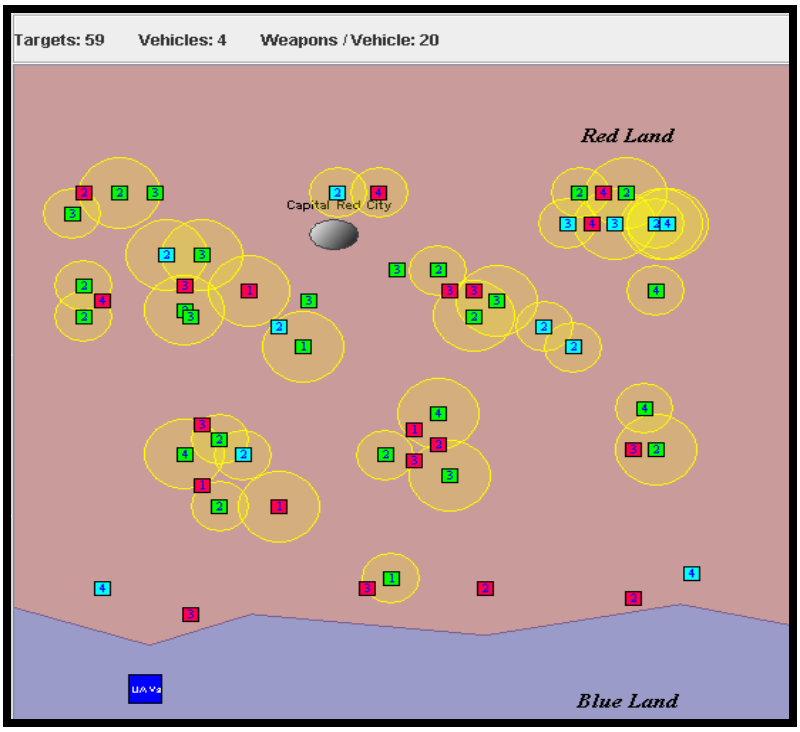


Figure 29: Training Scenario Used for All Tutorials

The users had access to all relevant training material throughout the experiment and were encouraged to ask questions at any time during the experiment.

Since the order of the experimental levels was randomized, participants received additional training and a tutorial before beginning a level with new TBD tools. For example, if a participant was slated to use the Level 2 TBD interface during the first scenario, he would train on Level 1 and then Level 2 before he began the scenario. He would train on Level 1 first because Level 2 builds upon the tools used in Level 1. The completion of Level 2 training completed training for the remainder of the experiment. The participants began each scenario only after they stated that they understood the training and were ready to resume the experiment.

5.5 Hypotheses

The experimental hypotheses predicted that increasing levels of TBD would have the following effects:

- Improved understanding of the algorithm.
- Improved predictability of how weight inputs change the plan.
- Improved HMCDM plan quality.
- Improved dependability in the HGA testbed to generate quality solutions.
- Improved faith in the HGA testbed to cope with a large variety of scenarios.
- Increased trust in the HGA testbed.
- Improved confidence in the HMCDM plan.
- Increased time between searches and fewer plans generated.
- Improved plan rankings in the fourth scenario.
- Improved decision-making support.

In addition, the following behavior was predicted to support the above effects:

- The sensitivity analysis in Level 2 will prevent users from not changing the coefficients sufficiently to generate new plans. Users in Levels 0 and 1 will have trouble knowing how much to change the coefficients.
- The Sliding Bars will be reported as the easiest method to physically input the weights, as opposed to the Pie Chart Control. The Sliding Bars can be manipulated only with the mouse. The Pie Chart Control requires the use of a mouse and keyboard, which may make changing the weights more time consuming than using the Sliding Bars.
- Participants will develop a search strategy in Level 2 due to the Weight and Metric Histories. By visualizing how past weight inputs affect plan metric outputs, users can develop their own search heuristics to explore the solution space. An example heuristic is provided in Figure 20.
- HMCDM plans will be preferred to global plans because the user has control over the entire solution process, and not just the initial definition of coefficients.

- The HGA will be preferred to the global search algorithm because of the human-computer collaborative methods improving the quality of its solutions.
- The quality of the global plan will be independent of the level of HGA TBD, but will be dependent on the number of scenarios the participant completed. As participants gain experience selecting coefficients for the global search, the quality of global plans will improve.
- Due to the variability in the target locations, users will build slack into the plan metrics to ensure those metrics meet constraints defined in each scenario.

It is necessary to note that the purpose of TBD is not to increase trust, but to improve the user's ability to develop appropriate trust in the HGA. This means that the user can correctly evaluate trust in his own ability and in the individual computerized components at specific times. The scenarios were designed so that the models used by the HGA could be assumed capable of generating quality solutions. Each scenario explained that there might be some variability in the metric calculations, but this variability was consistent over all experimental levels.

The TBD tools were designed for the user to better evaluate his/her performance, understand the nature of the problem and the methods being used to solve it, and increase his/her ability to steer the HGA to higher quality solutions. This increased system transparency should improve the user's calibration of *appropriate* trust in the computer components. However, due to the aforementioned improvements and the assumption that the HGA is capable of generating quality solutions for the given scenarios, it is hypothesized that the participants will increasingly trust the HGA as more TBD tools are added to the interface. It is therefore hypothesized that levels of the experiment that do not provide the benefits of increased situational awareness and HGA controllability are less likely to be trusted than levels that provide this information and capability.

5.6 Data Collection

The data captured during the experiment included all the HGA generated plans, their corresponding weights, and the order of generation. The global plans and their corresponding weights were saved, but all of the plans considered by the global search were not saved. The selection of HMCDM or global plan after each scenario, and the scenario 4 rankings were

recorded. Participants completed questionnaires consisting of written responses, yes or no questions, and 3 and 7 point Likert scales. All questions found in the questionnaires can be found in Appendix A.

The questions used in the questionnaire were developed from Rempel's three levels of trust, predictability, dependability, and faith. Trust was directly evaluated in the HGA testbed using the Lee and See definition of trust. The user's confidence in the final plan was also evaluated due to the importance of self-confidence in developing trust.

5.7 Methods of Data Analysis

Statistical calculations were performed with SPSS 14.0. In addition to evaluating various plots and descriptive data, the following tests were performed on the appropriate data:

- ANOVA: Parametric test to compare the distribution of two or more variables. This test was performed on data pertaining to the number of plans generated. A significant p-value concludes that at least one variable differs from another variable.
- Binomial test: Compares binary data to the binomial distribution. This test was performed on survey responses. A significant p-value concludes that the data was not from a binomial distribution with a specified probability of success.
- Cochran's Q test: Non-parametric test used to compare the distributions of two or more related binary variables. This test was performed on survey data. A significant p-value concludes that the at least one variable differs from another variable.
- Friedman test: Non-parametric test used to compare the distribution of two or more related variables. This test was performed on Likert 3 and 7-point scales and scenario 4 plan ranking data. A significant p-value concludes that at least one variable differs from another variable.
- McNemar test: Non-parametric test used to compare the distributions of two related binary variables. This test was performed on binary data. A significant p-value concludes that the two variables are different.
- Paired Samples T-test: Parametric test to compare the distribution of two dependent variables. This test was performed on data pertaining to the number of plans generated. A significant p-value concludes that the two variables are different.

- Wilcoxon Signed-Rank Test: Non-parametric test used to compare the distributions of two related variables. This test was performed on Likert scale data and scenario 4 plan ranking data. A significant p-value concludes that the two variables are different.

In addition to the quantitative analysis, a qualitative analysis was performed, including interpreting graphical data and evaluating the written questionnaire free responses. The following graphical aids were used in the qualitative analysis: Draper Laboratory Decision Space Visualization (DSV), MATLAB, and Microsoft Excel.

Chapter 6

Experimental Results and Discussion

This chapter explains the results obtained from the TBD HGA experiment. Questionnaire results are presented to show the effects of the TBD tools on the participants and their development of trust. Also, plan quality and plan generation efficiency are compared across the levels of TBD. Using survey responses and plots of the solution space, the participants' strategies are explored. Finally, a discussion highlights conclusions drawn from the experiment.

6.1 Questionnaire Results

The user's ability to guide the HGA, user understanding, and operating difficulty are explained in this section. The TBD tools and various measures of trust are also presented. The alpha value for significance is .05. P-values between .05 and .10 are defined as showing non-significant trends approaching significant, because only a small sample size was used. P-values between .10 and .15 are defined as showing a non-significant trend that might be approaching significance.

6.1.1 User Ability to Guide Algorithm

The user's ability to guide the algorithm was measured with the following question:

- Rate how the HGA testbed helped support your ability to guide the algorithm to produce the plans you wanted:

1	2	3	4	5	6	7
Very Unhelpful	Unhelpful	Fairly Unhelpful	Neutral	Fairly Helpful	Helpful	Very Helpful

A box plot of the results is shown in Figure 30. There is a non-significant trend that might be approaching significance that the increased levels of TBD led to increased support for the user's ability to guide the algorithm (Friedman p-value = .142). The Level 0, 1, and 2 median ratings were respectively "Fairly Helpful", "Fairly Helpful-Helpful", and "Helpful." There is a non-significant trend approaching significance that Level 2 was more supportive than Level 0 (Wilcoxon p-value = .066). This is likely the result of the sensitivity analysis provided in Level 2.

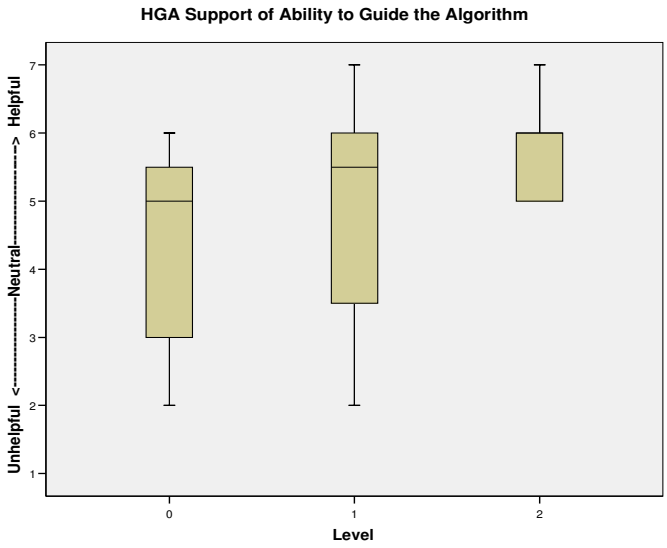


Figure 30: Box Plot - HGA Support of Ability to Guide Algorithm

6.1.2 User Understanding

The user's understanding of the algorithm logic was measured with the following question:

- Rate how the HGA testbed helped support your understanding of the algorithm's logic (understanding of how algorithm works, what it does, why it does it, etc.):

1	2	3	4	5	6	7
Very Unhelpful	Unhelpful	Fairly Unhelpful	Neutral	Fairly Helpful	Helpful	Very Helpful

A box plot of the results is shown in Figure 31. There is a non-significant trend approaching significance that incorporating TBD (Levels 1 and 2) helped support the user's understanding of the algorithm logic (Friedman p-value = .068). The Level 1 and 2 median ratings were "Fairly Helpful-Helpful" and the Level 0 median rating was "Fairly Helpful." The difference between Level 0 and Levels 1 and 2 might be approaching significance (Wilcoxon p-values: Level(0-1) = .102, Level(0-2) = .066).

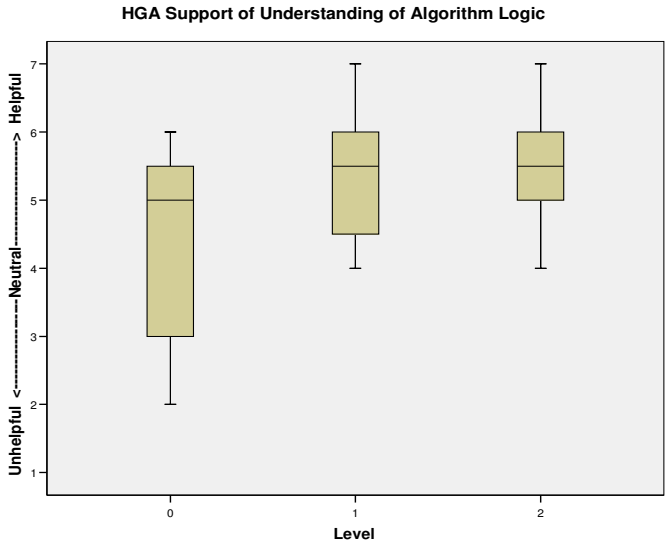


Figure 31: Box Plot - HGA Support of Understanding of Algorithm Logic

6.1.3 Operating Difficulty

The operating difficulty of the HGA was measured with the following question:

- How would you rate the difficulty in operating the HGA testbed?

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

A box plot of the results is shown in Figure 32. There is not enough evidence to conclude that the increased levels of TBD affected the difficulty of operating the HGA testbed (Friedman p-value = .705). The median difficulty ratings for all levels were "Easy."

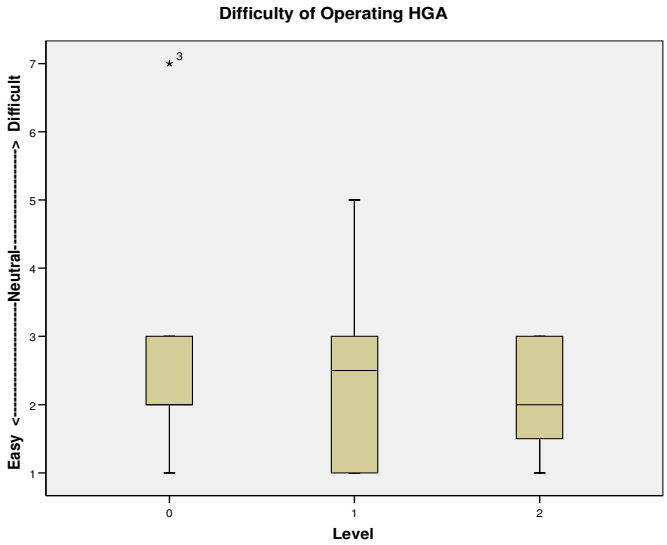


Figure 32: Box Plot - Difficulty of Operating HGA

6.1.4 Trust-Based Design Tools

6.1.4.1 Controls: Sliding Bars, Pie Chart, Pie Chart with Sensitivity Analysis

Various aspects of the HGA controls were evaluated: difficulty of use, effect on search strategy, effect on understanding, and effect on decision making.

The difficulty of operating the HGA was measured with the following question:

- Rate how difficult it was to use the TBD control:

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

A box plot of the results is shown in Figure 33. The difficulty of operating the Sliding Bars and Pie Chart Control both had a median rating of "Fairly Easy-Easy." The Pie Chart Control with Sensitivity Analysis Display was rated "Easy", most likely because the Sensitivity Analysis Display provided information to help the user select appropriate weights. However, the difference between the difficulty of operating the controls was not significant (Friedman p-value = .846) due to variability in the rankings.

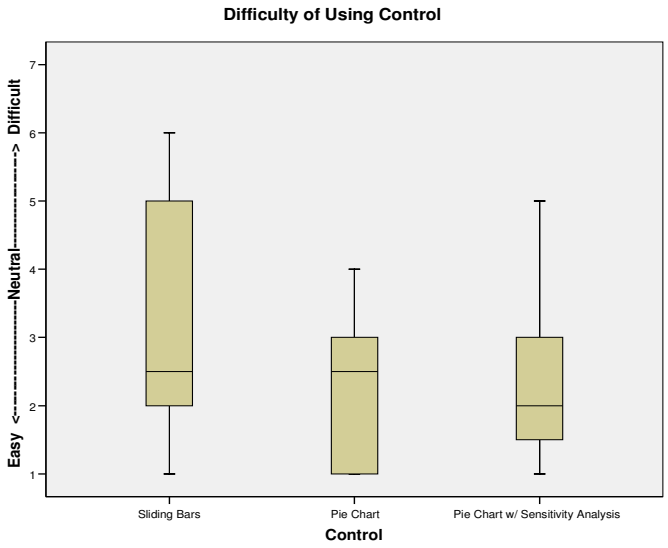


Figure 33: Box Plot - Difficulty of Using Control

Seven of the 8 participants (87.5%) commented that the Sliding Bars had an effect on their strategy. One user commented that the Sliding Bars were the fastest method of changing

the weights, while another said that they were slower than the other controls. Another commented that they were harder to fine tune. A participant who used the Sliding Bars during the first scenario said, "*It was very helpful to be able to "play around" with what is important*." Only two users commented that the Pie Chart Control had an affect on their strategy. One user, who said it affected his strategy, commented that the Pie Chart Control allowed him to "*easily change values in ratio to another*." Another user, who said the Pie Chart Control did not affect his strategy, said that the Pie Chart Control was easier to implement. The Pie Chart Control with Sensitivity Analysis affected the strategy of 6 of the 8 participants (75%). Five of the 6 participants explained why the Pie Chart Control with Sensitivity Analysis Display affected their strategy:

- *"It enhanced my strategy, made things more efficient."*
- *"Yes, instead of guessing and checking to try and change solution, I got a new one every time."*
- *"Could pick weightings much better."*
- *"Yes, able to see slightly better solutions towards end, when last time everything seemed to get too settled to affect."*
- *"Yes - it helped me know which constraints to change."*

The pie chart graphic and renormalizing the weights before the search did not have a significant effect on search strategy. The sensitivity analysis had a positive strategy effect on the majority of the participants. The Cochran's Q test confirms that the type of HGA control had a significant effect on whether the participants reported that the control affected their search strategy (p-value = .015).

The HGA control also affected the user's understanding of the algorithm's logic (Friedman p-value = .032). For Level 0, there was one positive comment about the Sliding Bars' effect on understanding the algorithm logic, one neutral comment, and five negative comments. The majority of the negative comments indicate that the participants did not know how far to move the Sliding Bars to get a new plan in the next search. For Level 1, there were five positive comments and three neutral comments about the Pie Chart Control. One user commented that the change in plan metrics usually correlated to the change in weights. The remaining comments indicated that participants liked that the weights summed to one or that the graphic improved

how they visualized the weights. One user commented that the Pie Chart Control *"helped me see relative increases/decreases that the program would see."* For Level 2, there were seven positive comments and one neutral comment about the Pie Chart Control with Sensitivity Analysis. The positive comments indicated that the users liked knowing how much they needed to change the weights to get a new plan. The one neutral comment was that it was difficult to understand how the sensitivity analysis corresponded to changing more than one weight. Because the sensitivity analysis was only valid for changing one weight between searches, the participant was correct in pointing out that it was unclear what would happen if more than one weight were changed.

The control's support of decision making was evaluated with the following question:

- Rate how useful the TBD tool was in supporting your decision making:

1	2	3	4	5	6	7
Very Useless	Useless	Fairly Useless	Neutral	Fairly Useful	Useful	Very Useful

A box plot of the results is shown in Figure 34. The Sliding Bars, Pie Chart Control, and Pie Chart Control with Sensitivity Analysis had median ratings of "Neutral-Fairly Useful", "Neutral", and "Useful", respectively. Due to variability in the ratings, there is not enough evidence to conclude there is a difference in their ability to support decision making (Friedman p-value = .435).

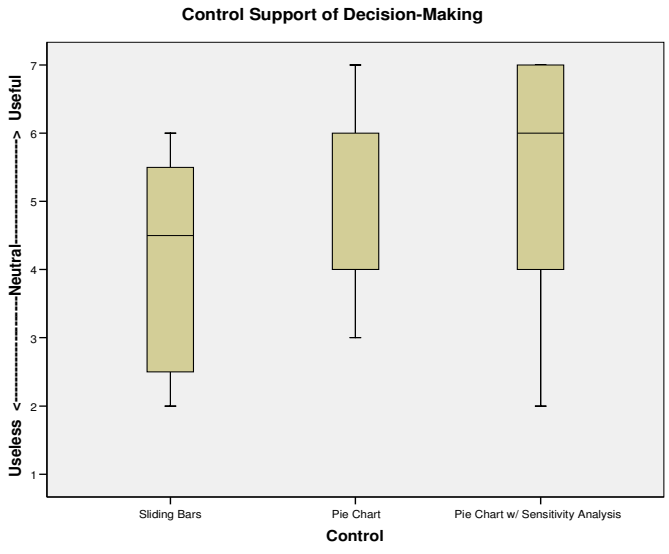


Figure 34: Box Plot - Control Support of Decision Making

The participants were asked how they used the controls to generate new plans. Comments about the Sliding Bars indicate the effects they had on the participants:

- *"Rapidly positioned one at a time to drive desired metric in desired direction."*
- *"I usually modified one at a time to see if that particular constraint would improve results."*
- *"Made many large adjustments, hard to fine tune. I did a lot of swinging from end to end to ensure change happened."*
- *"With the mouse, arrow keys would be better for fine tuning."*
- *"Guess and check to slide how far and get a new solution."*
- *"I used them from 1-10 to get different ratios."*
- *"Changed one bar at a time (but sometimes changed more than one at a time), made drastic changes if kept getting the same results."*
- *"It allowed me to isolate and focus on the effect of one variable."*

The comments indicate that some participants made drastic changes to the sliders to ensure a new plan would result in the next search. This was likely because the Sliding Bars were difficult to fine tine to specific numerical values and it was unknown how far they needed to be moved to get a new plan. Three of the users indicated that they only changed one metric at a time between searches. The final comment, "*it allowed me to isolate and focus on the effect of one variable,"* shows how the Sliding Bars give the deceptive impression that changing one weight has no impact on the remaining weights. This participant preferred the Sliding Bars to the Pie Chart Control because of this misleading quality.

In Level 1, four of the participants commented that they did not consider the pie chart graphic when picking their weights; they just looked at the normalized weight values. This indicates that the Pie Chart Control can also accommodate users who are not visually inclined. The four remaining comments described how the participants used the Pie Chart Control and some of its positive attributes:

- *"Gave relative weights and let chart normalize values with adjust function."*
- *"Extremely easy in adjusting weights in small amounts relative to each other."*
- *"Liked typing in weights better than sliders."*
- *"Visualized the normalized risks."*

In Level 2, the majority of the comments focus on the sensitivity analysis:

- *"I used the sensitivity extensively so that I only ran plans that I knew would result in path changes."*
- *"Tried to switch time, risk outside of no change range."*
- *"Looked at generalized picture of pie chart (that was confusing), but the numerical sensitivity provided great ranges for the data."*
- *"The sensitivity analysis is new to me so I didn't really understand it. The weightings were better than the sliders."*
- *"Allowed me to see how much adjustment was necessary to cause change."*
- *"The values for which a change would occur made it very helpful."*
- *"It was available, and I used it to input values. Primarily I focused on plan metrics to determine my changes though. The only sensitivity I used was the change range to determine whether a change was going to have an effect. This was my first scenario so I focused on learning how extremes affected my end goals of time and attrition. As stated above, I used it for inputting values and reading what changes would be effective."*
- *"Move one weight of interest outside of no change range to move score in desired direction."*

The comments indicate a stark difference in the operation of the Sliding Bars and the Pie Chart Control with Sensitivity Analysis Display. Instead of guess and check or drastic weight changes, most user inputs were guided by the sensitivity analysis. A drawback of the sensitivity analysis and its graphic was the initial difficulty to understand how to use correctly.

After the completion of all four scenarios, the participants were asked which tools they would prefer if they had to do another HGA search. Five of 8 selected the Pie Chart Control over the Sliding Bars. One of the participants who preferred the Sliding Bars had difficulty

entering numbers in the excel spreadsheet when operating the Pie Chart Control. Another participant, as mentioned above, believed the Sliding Bars allowed him to only change one weight at a time. Seven of 8 wanted sensitivity analysis to be provided to help steer their search.

6.1.4.2 Displays

The difficulty of using the TBD displays (Rationale Window, Plan Cycle Option, Metric History, Weight History) were evaluated with the following question:

- Rate how difficult it was to use the TBD display:

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

A box plot of the results is shown in Figure 35.

The displays' support of decision making was evaluated with the following question:

- Rate how useful the TBD display was in supporting your decision making:

1	2	3	4	5	6	7
Very Useless	Useless	Fairly Useless	Neutral	Fairly Useful	Useful	Very Useful

A box plot of the results is shown in Figure 36.

6.1.4.3 Rationale Window

Six of 8 participants used the Rationale Window during Level 1 and its median difficulty rating was "Easy." The following comments illustrate how it was used:

- *"Tracked changes in metrics to make decisions about changing weights."*
- *"I used it to compare current plan with previous plan."*
- *"Helped weigh pros/cons, especially when close at end."*
- *"To check the changes that had been made."*
- *"Check for +/-, a lot easier that comparing numbers, good for evaluating tradeoffs."*
- *"See if the scenario improved my stats that I was looking at."*

Five of 8 participants said that the Rationale Window affected their strategy by showing what changes were occurring. One user commented that the Rationale Window helped fine tune his strategy. Five of 8 also commented that the Rationale Window had a positive effect on their understanding of the algorithm's logic. Three of 8 said it had a neutral effect. One user commented that the Rationale Window, "*supported by relating weights in the pie chart to scenario numbers* (metrics)." The median utility rating was "Useful" which was the highest rating of all the other TBD displays. All 6 participants who used the Rationale Window in Level 1 selected it as a tool they would want for future HGA searches.

6.1.4.4 Plan Cycle Option

Four of 8 participants used the Plan Cycle Option during Level 1. Its median difficulty rating was "Fairly Easy-Easy." The comments indicate that the participants used it to watch the moves being made by the algorithm, but it had no effect on strategy. The median support of decision making rating was "Fairly Useless-Neutral." Three users said it improved their understanding of algorithm logic by showing the incremental changes to the plan and path/plans considered by the search algorithm. Five users commented that it did not affect their decision making. Three of 8 users, only one of which actually used it in Level 1, selected it as a tool they would want for future HGA searches.

6.1.4.5 Weight History

Five of 8 participants said they used the Weight History. Its median difficulty rating was "Easy." All 8 participants commented how they used it:

- *"Just sort of glanced at it to see trends."*
- *"Saw what I had done before."*
- *"Just looked at it to see where I had been."*
- *"To see the past choices."*
- *"I looked to see how I had previously weighed the constraints."*
- *"I didn't reference it at all."*
- *"Looked at general trends in weight history."*

- *"Glanced at it a couple of times to ensure not repeating self."*

The comments show that most participants briefly referenced the chart, even if they did not consider the information in their decision making. The last comment, "*glanced at it a couple of times to ensure not repeating self*" may indicate a misconception of the local search. Due to the nature of the local neighborhood search, repeating the same weights used to generate a previous plan does not guarantee that the same plan will be generated. It some cases, it may be a good strategy to repeat the same set of weights from different locations in the solution space, which is how the global search algorithm operates. Only 1 of 8 participants said that the Weight History affected his strategy. Two of 8 said it improved their understanding of algorithm logic. One of those users commented, "*It aided my understanding by helping visualize what causes a change*." Six of 8 comments were neutral. The median rating for how well the Weight History supported decision making was "Neutral." Four of 8 participants selected the Weight History if they needed to complete future HGA searches.

6.1.4.6 Metric History

Five of 8 participants said they used the Metric History. Its median difficulty rating was "Neutral-Fairly Easy." Six participants commented how they used it:

- *"Saw what had dropped to try and keep them there."*
- *"Could see the trends of the plans I generated."*
- *"Watched change trends."*
- *"I could see which scenarios yielded best results."*
- *"Trial and error to see how my attrition variable would change."*
- *"Tracked performance metrics as plan changed."*

Similar to the Weight History, comments about the Metric History indicate that most participants briefly referenced the chart to compare plans and metrics. Only 2 of 8 participants said that the Metric History affected their strategy. One of those participants said that it "*helped compare current plans with previous ones*." Three of 8 said it improved their understanding of the

algorithm. Five comments were neutral. The following are some of the comments about how the Metric History affected understanding of the algorithm:

- *"None of above, interesting to look at though."*
- *"Improved - helped provide comparison."*
- *"Supported but not extremely beneficial at my level of expertise."*
- *"Improved - immediate feedback on how "good" the plan was."*
- *"Didn't really seem to need it."*

The median rating for how well the Metric History supported decision making was "Neutral-Fairly Useful." Similar to the Weight History, 4 of 8 participants selected the Metric History if they needed to complete future HGA searches. One user commented that, "*the metric charts would have been interesting at the end. Not enough expertise initially to feel comfortable with them.*" Like this user, it may take experience with the HGA to understand the value of the information presented in the Weight and Metric Histories.

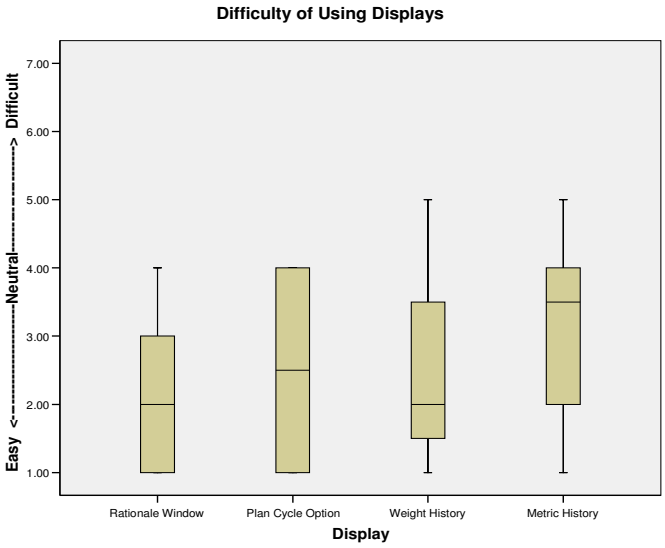


Figure 35: Box Plot - Difficulty of Using Displays

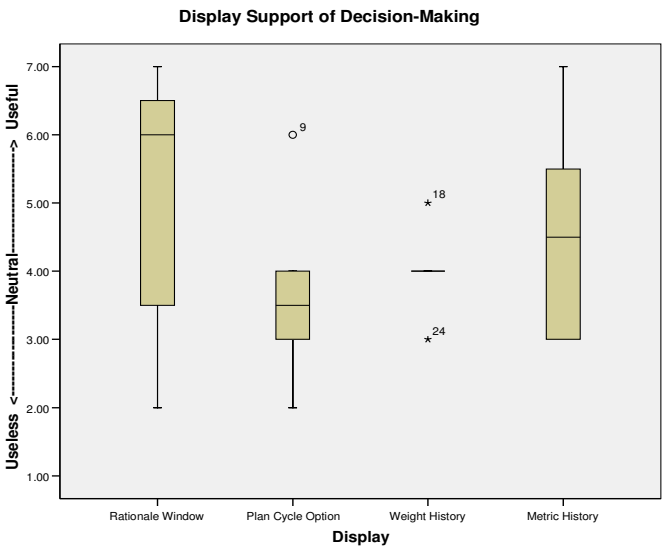


Figure 36: Box Plot - Display Support of Decision Making

6.1.5 Trust

Trust was evaluated by measuring predictability, dependability, faith, trust, confidence, and trust specificity and resolution.

6.1.5.1 Predictability

For a user to develop trust in the HGA, he must be able to predict how his inputs will affect the HGA and result in a new plan. The utility of the HGA rests on the assumption that the user can predict how his inputs will affect the HGA and can use those inputs to generated quality solutions. In the experiment, there is a significant trend that increasing levels of TBD helped the users to predict how their input weights change the current plan.

The user's ability to predict how weight inputs change the plan was measured with the following question:

- How would you rate the predictability of how the weights you input would change the current plan?

1	2	3	4	5	6	7
Very Unpredictable	Unpredictable	Fairly Unpredictable	Neutral	Fairly Predictable	Predictable	Very Predictable

A box plot of the results is shown in Figure 37. The median predictability ratings for Levels 0, 1, and 2 were respectively "Neutral-Fairly Predictable", "Fairly Predictable", and "Predictable." There is not enough evidence to conclude there is a difference between Levels 0 and 1, but there is a significant difference between Level 2 and the other levels (Wilcoxon p-values: Level (0-2) = .026, Level(1-2) = .039).

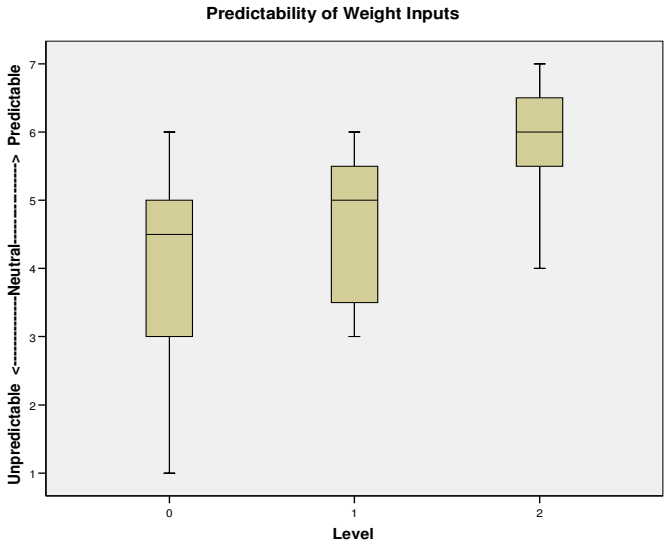


Figure 37: Box Plot - Predictability of Weight Inputs

Level 2 introduces the Sensitivity Analysis Display, which likely contributed to the significantly higher predictability rating. Since Level 1 was not significantly different from Level 0, the comparison of the background color of the weights on the Pie Chart Control to the metric changes on the Rationale Window, both introduced in Level 1, did not significantly improve predictability.

6.1.5.2 Dependability

How much the HGA could be depended on to generate "good enough" or optimal solutions was rated by the participants with the following question:

- To what extent can you depend on the HGA testbed to aid you in finding a good enough or optimal solution?

1	2	3	4	5	6	7
Very Undependable	Undependable	Fairly Undependable	Neutral	Fairly Dependable	Dependable	Very Dependable

A box plot of the results is show in Figure 38. Increasing levels of TBD show increasing median dependability ratings of the HGA testbed, but variability in the ratings makes the trend insignificant. The median dependability ratings ranged from "Fairly Dependable" to "Fairly Dependable-Dependable."

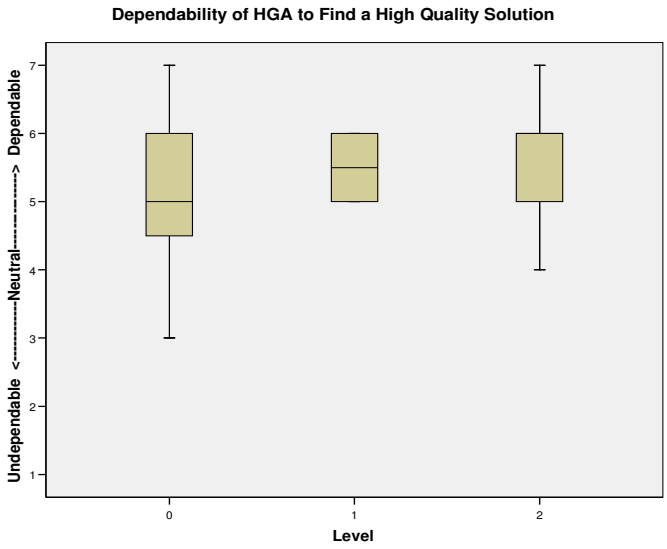


Figure 38: Box Plot - Dependability of HGA to Find a High-Quality Solution

6.1.5.3 Faith

Faith in the HGA was measured with the following question:

- What degree of faith do you have that the HGA testbed will be able to cope with a large variety of scenarios?

1	2	3	4	5	6	7
Very Unfaithful	Unfaithful	Fairly Unfaithful	Neutral	Fairly Faithful	Faithful	Very Faithful

A box plot of the results is shown in Figure 39. There is a significance trend that incorporating TBD, Levels 1 and 2, improved the user's faith in the HGA testbed to cope with a large variety of scenarios (Friedman p-value = .047). The median rating for Level 0 was "Fairly Faithful" while Level 1 and 2 ratings were both "Faithful." There is not enough evidence to conclude there is a difference between Levels 1 and 2, but it is nearly significant that Level 0 is different from Levels 1 and 2 (Wilcoxon p-values: Level (0-1) = .059, Level(0-2) = .059).

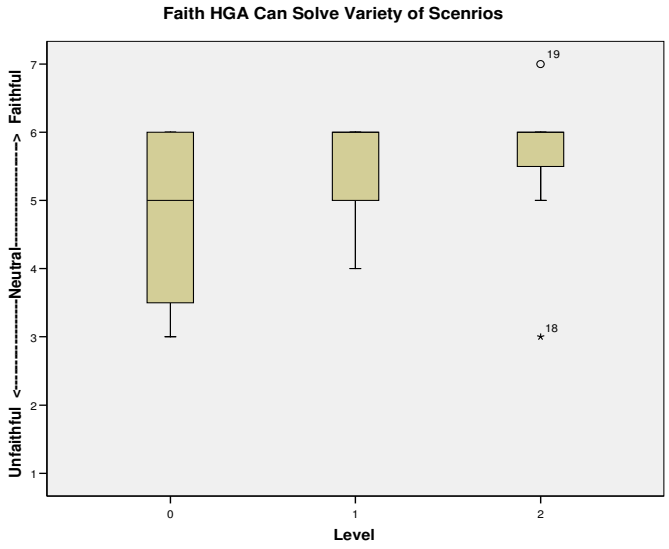


Figure 39: Box Plot - Faith HGA Can Solve a Variety of Scenarios

6.1.5.4 Trust

Trust was measured with the following question:

- How much do you trust the HGA testbed?

(Trust: The attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability)

1	2	3	4	5	6	7
Very Untrustworthy	Untrustworthy	Fairly untrustworthy	Neutral	Fairly Trustworthy	Trustworthy	Very Trustworthy

A box plot of the results is shown in Figure 40. There is a trend approaching significance that incorporating TBD (Levels 1 and 2) improved the user's trust in the HGA testbed (Friedman p-value = .074). The Level 0, 1, and 2 median rating were respectively "Fairly Trustworthy", "Fairly Trustworthy-Trustworthy", and "Trustworthy." There is not enough evidence to conclude there is a difference between Levels 1 and 2, but it is nearly significant that Level 0 is different from Levels 1 and 2 (Wilcoxon p-values: Level (0-1) = .102, Level(0-2) = .059).

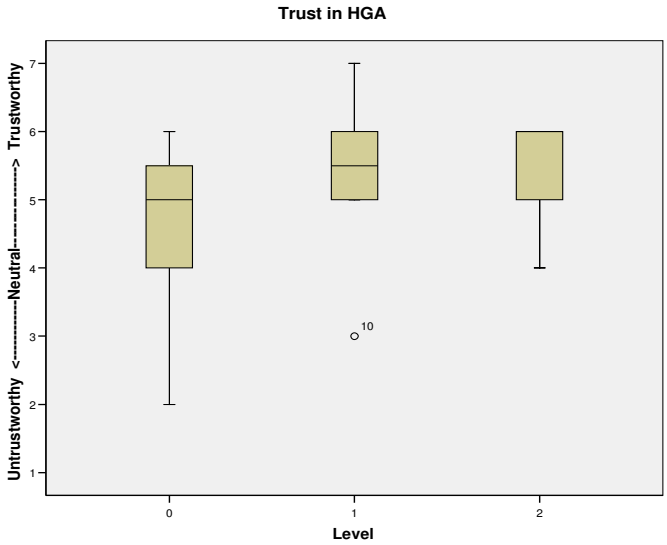


Figure 40: Box Plot - Trust in HGA

6.1.5.5 Confidence

Confidence in the HMCDM plan was measured with the following question:

- How much confidence do you have in the quality of your final plan, with respect to the mission description?

1	2	3	4	5	6	7
Very Unconfident	Unconfident	Fairly unconfident	Neutral	Fairly Confident	Confident	Very Confident

A box plot of the results is shown in Figure 41. There is a trend that may be approaching significance that incorporating TBD (Levels 1 and 2) improved the user's confidence in the final plan (Friedman p-value = .143). The Level 0, median rating was "Fairly Confident" and Level 1 and 2 ratings were both "Fairly Confident-Confident." There is not enough evidence to conclude there is a difference between Levels 1 and 2, but it is nearly significant that Level 0 is different from Levels 1 and 2 (Wilcoxon p-values: Level (0-1) = .102, Level(0-2) = .096).

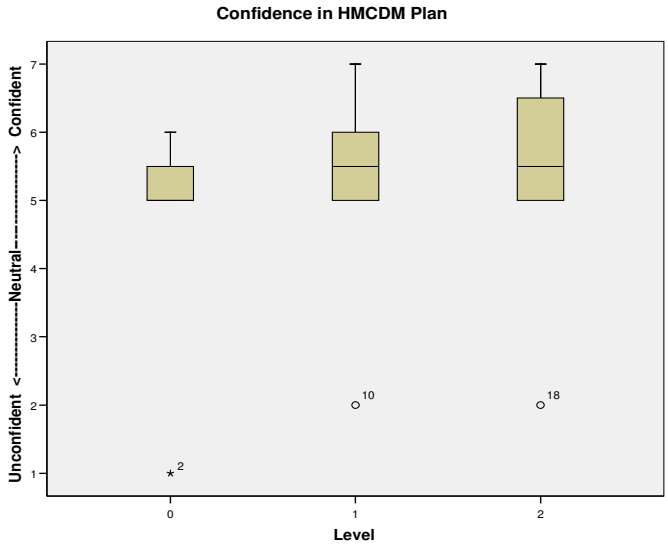


Figure 41: Box Plot - Confidence in HMCDM Plan

6.1.5.6 Specificity and Resolution

In an attempt to glean the participants' specificity and resolution of trust, the following question was asked:

- Were there system components (controls, windows, metrics, algorithm, etc.) that you trusted more than others? Which ones? Why? Was there anything that caused confusion, lack of trust, etc?

It is interesting to note that the majority of the users answered this question by commenting about the TBD controls, especially the Sensitivity Analysis Display:

- *"Sliding bars - hard to get an exact number, pie chart w/sensitivity analysis: liked this the best."*
- *"Pie chart with sensitivity analysis was real nice because you knew how much to change to get a different result. I didn't like the sliding bars, they gave no information."*
- *"Level 2 was the best by far. It made it much easier to generate new optimums rather than staying at the same plan. In level 1, I generated the same plan even though I changed values several times. The sliding bars were the trickiest. No easy method to compare incremental plans. I trusted level 2 the most."*
- *"Didn't like sliding bars, pie chart was good (not the actual pie chart), rationale window was good, very useful, sensitivity analysis was very good, histories weren't very useful."*
- *"Not an issue of trust, found sliding bars and 2 history charts not useful. When just sliding bars I became a bit frustrated with my eventual lack* [of ability] *to change* [the] *plan with most priority* (weight) *changes."*
- *"The metric charts would have been interesting at the end. Not enough expertise initially to feel comfortable with them."*
- *"Sliding bars - very fast, but exact placement is hard and no indication of no change range, sensitivity analysis - slow speed offset by utility and no wasted time with no change* [in the plan] *from changed weights"*

These comments correspond and support the analysis performed on the TBD tools. It appears that most participants associated trust with how they controlled the HGA and how well the HGA supported their ability to intelligently steer the algorithm.

6.1.6 Trust Information

In the post experiment questionnaire, participants were asked about their trust of the data and HGA calculations. The participants were asked if they would change their method of selecting their final plan if there was a high probability that the expected value calculation was wrong. Seven of 8 said that this would affect how they selected their final plans (binomial p-value .07). How much it would affect their plan, some users commented, depended on how heavily they weighted expected value in their decision making. Those that looked at total value and not expected value would not have been as affected as those that looked primarily at expected value. A few users commented that if the expected value calculation was wrong, they would weight other metrics as more important in their decision making.

Seven of 8 participants said that knowing about variability in the data would change how they selected their final plan. The users' comments reveal their attitude towards knowledge of data variability and how they would compensate for it:

- *"Yes, would maybe not push the limits so much (i.e. more cushion for mission time)."*
- *"Yes, I would get conservative and only go after close targets where help can be maximized."*
- *"Yes, I wouldn't have been as strict in meeting the requirements."*
- *"Perhaps, but variabilities have a tendency to cancel out. I may leave more margin though and go conservative."*
- *"I would probably include more factors of safety…"*
- *"Yes - survivability would become more important because it's not worth risk of loss or possible wrong target."*
- *"No, this is the best information I have to act on, so it is what I would rely on."*
- *"Would likely weigh each criteria differently, but not drastically - variability means trust in data is shaky, but no better solution exists at that time."*

The comments indicate that data variability may change how the user evaluates quantitative metrics and qualitative aspects of the plan, which will affect how the user steers the algorithm and generates plans. It also reveals the different attitudes users have towards incorporating trust information into their planning. The majority of the participants would add slack to their plan to improve its likelihood of success. A few would not add slack by assuming they could not increase the robustness of the plan.

Each experimental scenario explained that there was variability in the target locations that could affect the accuracy of all the metrics. The participants were asked how variability in the target locations affected their strategy of selecting a quality plan. Surprisingly, only 1 of the 8 accounted for the variability by adding slack into the plan to ensure it met the 2-hour time constraint. One user commented that the variability "*was a limitation that couldn't be helped so the best solution was determined from available data.*"

When asked if knowing the magnitude of data inaccuracy or variability would affect decision making, all eight participants answered affirmatively. Most comments were similar to the comments above about knowledge of data variability. Here are some additional comments:

- *"Yes - higher variability/standard deviation would give quantitative view into how good the predictions were."*
- *"Yes - as variability increases, attrition also becomes more important, it's only worth the risk when you know for certain what you hit."*
- *"Yes, helps weigh inaccuracies against each other."*

The above results indicate that providing the user with trust information will affect how the HGA is used to generate plans, which will determine the types of plans generated. The fact that 7 of 8 users said that knowledge of data variability would affect their decision making, but only 1 of 8 considered it in the experiment, indicates that users need help incorporating this information into their decision making. As revealed by a few user comments, robust planning may not always be intuitive. Providing trust information to help the user could be beneficial to generating plans robust against known uncertainty. All 8 users said that knowing the magnitude of the inaccuracy would be helpful to their decision making.

6.2 HGA Effectiveness

The effectiveness of the HGA was measured by its efficiency in generating solutions and the quality of those solutions. Measurements of search efficiency include the total number of searches performed, the number of unique plans generated, and the percent of searches that resulted in unique plans. Participants compared the quality of their HGA solutions to global solutions and rank ordered all of the experimental solutions in scenario 4.

6.2.1 Search Efficiency

Increasing levels of TBD significantly reduced the number of searches performed during the 15-minute time limit (ANOVA p-value = 0). A box plot of the data is show in Figure 42. The mean number of searches performed in Levels 0, 1, and 2 were respectively 40.4, 22.5, and 8.25. There are two reasons why Level 1 might have reduced the number of searches compared to Level 0. First, Level 1 added the Rationale Window to the display, which included new information for the user to interpret. Second, the Pie Chart Control, as some users commented, took more time to set the weights than the Sliding Bars. Comparing the number of searches in Level 1 to Level 2, the sensitivity analysis in Level 2 required 10-30 seconds or more to calculate. During this time, the participant could interpret data presented on the Weight and Metric History graphs, but had to wait until the sensitivity analysis was complete to enter new weights and begin another search. The Sensitivity Analysis Display greatly reduced the number of searches performed.

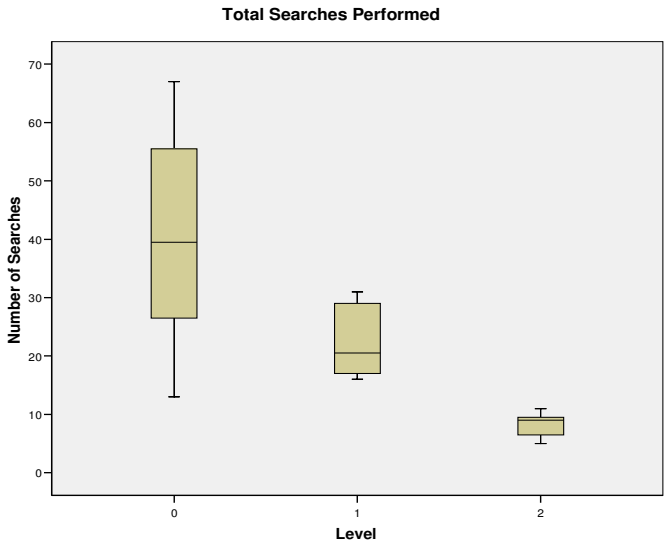


Figure 42: Box Plot - Total Searches Performed

Not all searches resulted in unique plans. A plan is considered unique if it is different from the previous plan generated. It is possible for plans to be repeats and still be unique as long as a different plan was generated between them. The measure of unique plans measure how successful the user was at escaping the local optimum during each search. If the weights were not changed enough between searches, the same plan would remain the local optimum. Some of the plots in the Operator Strategies section show users intentionally returning to the same plan during their search. There is a significant trend that increasing levels of TBD decreased the number of unique plans generated (Friedman p-value = .007). A box plot of the results is shown in Figure 43. The mean number of unique plans generated in Levels 0, 1, and 2 were respectively 16.0, 8.9, and 6.75. The difference between the number of unique plans generated in Level 0 and Level 2 is significant (T-test p-value: .008). The differences between Level 1 and the other levels are nearly significant (T-test p-values: Level(0-1) = .057, Level(1-2) = .097). The lower levels of TBD had the greatest number of unique plans generated.

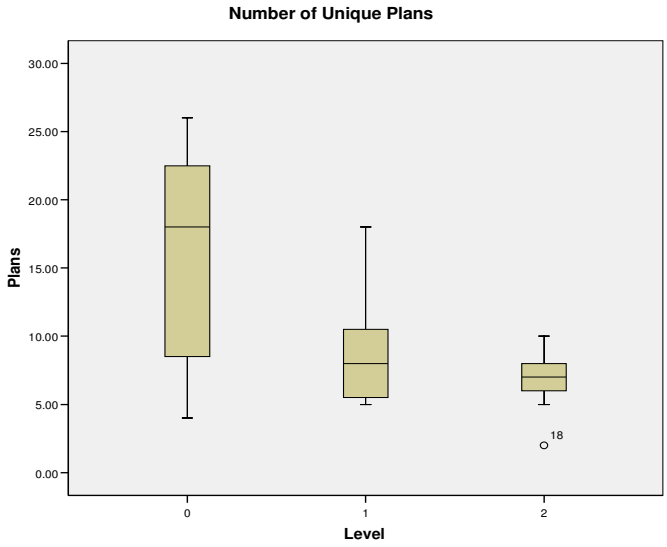


Figure 43: Box Plot - Number of Unique Plans

The percent of unique plans generated yields the best representation of the user's search efficiency, with respect to the number of attempts made to escape the locally optimal plan. A box plot of the results is show in Figure 44. The average percent of unique plans for Level 2 was 82%. The 18% of non-unique plans resulted from the participants' selection of weights and not due to errors in the Sensitivity Analysis Display. The average percent for Levels 0 and 1 were both approximately 39%. Level 2 significantly increased the percentage of unique plans generated by the user (T-test p-values: Level(0-2) = .001, Level(1-2) = .003). Because the percentage of unique plans was the same for Levels 0 and 1, the Pie Chart Control did not aid users in escaping the local optimum. The Sensitivity Analysis Display had its intended effect of helping the users escape the locally optimal plan. However, the benefit of a higher percentage of unique plans came at a computational cost that resulted in fewer searches and therefore fewer unique plans.

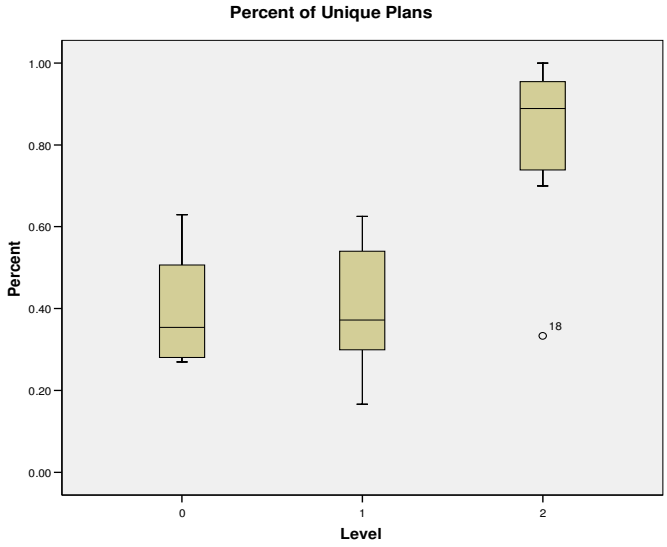


Figure 44: Box Plot - Percent of Unique Plans

6.2.2 *Solution Quality*

The solution quality of the HGA at its different experimental levels was measured against the solution quality of the global search algorithm. At the end of each scenario, when the user selected the global or HMCDM plan, the HMCDM plan prevailed 16 of the 24 times (67%) over the entire experiment. However, this result is not significantly different from flipping a fair coin

to decide between the HMCDM and global plan (binomial p-value = .152). Comparing the experimental levels, 7 of the 8 (87.5%) selected plans in Level 1 were HMCDM plans. Level 1 was nearly significantly different from flipping a fair coin (Binomial p-value = .07). The percent of HMCDM plans selected in Levels 0 and 2 were respectively 50% and 62.5%. There is not a significant difference in selection of the HMCDM plan or global plan across the TBD levels of the experiment (Cochran's Q-test = .311).

It is important to understand how most of the participants decided between the HMCDM and global plan. As discussed in the Operator Strategies section, most users looked solely at the plan metrics when making their decision. They selected the plan with the higher value or expected value that was within the time and attrition constraints. Plans that violated the constraints were either not considered or justified by the higher value they attained. Because the participants knew which plan they had generated with the HMCDM search and which was generated with their coefficients for the global search, there might have been bias in their selection of the better plan.

To account for this possible bias, the plan rankings from scenario 4 were evaluated. The participant did not know how the plans in scenario 4 were generated or that they had seen them before, so the user selected the plans based on their quality, and not their method of generation. For each level, the user's decision of HMCDM or global plan was compared to how they ranked those plans in scenario 4. Over all 24 decisions, there were only 2 discrepancies. Both discrepancies occurred when users in Level 0 selected the global plan, but ranked the HMCDM plan higher than the global plan in scenario 4. This changed the percent of HMCDM plans selected in Level 0 to 75%, but this result is still not significantly different from flipping a fair coin (Binomial p-value = .289). Using the scenario 4 rankings, it is significant that the HMCDM plan was preferred to the global plan over the entire experiment (Binomial p-value = .023). This result suggests that the HMCDM plans were superior to the global plans when the bias of how the plans were generated was removed. However, there is still not a significant difference in selection of the HMCDM plan across the TBD levels of the experiment (Cochran's Q-test = .549).

Using the McNemar test, there is not enough evidence to conclude there is a significant difference between the decisions made in Level 0 and the scenario 4 ranking results (p-value = .50). The two discrepancies may have been the result of shifting objectives and methods of

evaluating plan quality, the difficulty of ranking six plans, and fatigue from the prior 2-2.5 hours of the experiment. These results show consistency in the participants' judgment and no evidence of bias in how they evaluated plan quality. Therefore, evidence that the HMCDM plans were superior to the global plans according to the scenario 4 rankings must be interpreted cautiously.

After all experimental levels were completed, the participants were asked if they would rather use the global search or HGA, if they were required to generate plans for another scenario. Seven of 8 participants (87.5%) chose the HGA over the global search, if they needed to complete future searches, which is nearly significantly different from flipping a fair coin (Binomial p-value = .07). The users preferred the HGA (a few specified HGA with the appropriate tools) because it allowed them more control over the plan and they felt confident in their ability to generate plans to meet the mission requirements. The one user, Participant 2, who chose the global search, felt that the computer produced better plans than he did; however, as noted in the Operator Strategies section, this may not have been the case. One user commented that his opinion could change to favor the global search if the global search allowed him to input constraints rather than just inputting weights.

The scenario 4 rankings were also used to compare plan rankings across the levels of TBD and across the user's experience with the global search algorithm. The rankings are shown with a stacked bar chart in Figure 45 and with a box plot in Figure 46. The three TBD levels were used to categorize the HMCDM plans. The global plans were grouped into categories depending on when they were generated during the experiment: first, second, or third global search conducted by the participant. For example, Global 2 is the plan generated in the second scenario and was the second time the participant had used the global search algorithm.

There is not enough evidence to conclude there is a difference between the rankings of the HMCDM plans across the levels of TBD (Friedman p-value = .223). However, it is nearly significance that Level 1 HMCDM plan rankings differed from those of the other TBD levels (Wilcoxon p-values: Level(0-1) = .088, Level(1-2) = .078). The median ranking for Level 1 plans was 2, while Levels 0 and 2 had median rankings of 3.5.

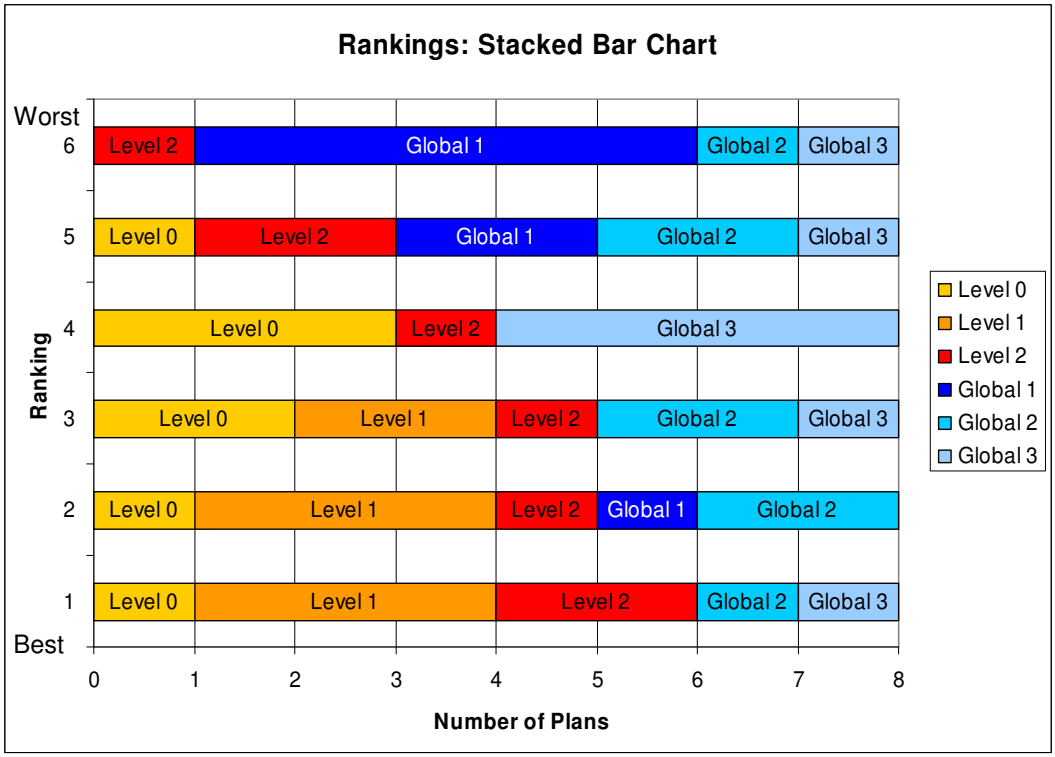


Figure 45: Stacked Bar Chart - Scenario 4 Rankings
(By level of TBD for HMCDM plans, and order generated for global plans).

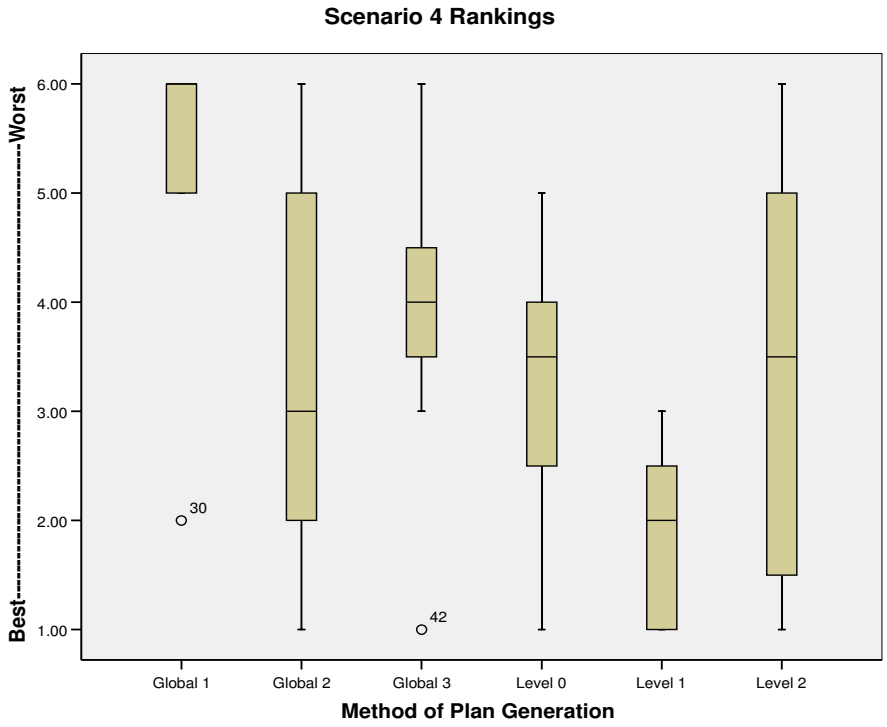


Figure 46: Box Plot - Scenario 4 Rankings
(By level of TBD for HMCDM plans, and order generated for global plans)

There is a nearly significant difference between the global plan rankings based on the user's experience with the global search algorithm (Friedman p-value = .072). It may be approaching significance that the first global search plans differed from the second and third global search plans (Wilcoxon p-values: Global(1st-2nd) = .106, Global(1st-3rd) = .120). The first global plan (Global 1) had a median ranking of 6 out of the 6 plans ranked. This indicated that the participants' first selection of global coefficients resulted in poor plans. The users learned from their first global search and from operating the HGA in their first scenario how to improve their selection of coefficients in the remaining global searches. However, the improvement did not continue after the second global search. The median rankings for the second global plan (Global 2) improved to 3, but the third global search plan (Global 3) increased to 4. This may be explained by the fact that the participants did not know that all the scenarios were identical and therefore were unaware that incorporating previous search experience would be beneficial to the next global search.

6.3 Operator Strategies

Operator strategies are evaluated from questionnaire responses and three types of plots are used to show unique aspects of the solution space.

6.3.1 General Search Strategies

Evaluation of the questionnaire response to the following question yielded two apparent search strategies implemented by the participants over the entire experiment:

- How would you describe your strategy for finding the best plan?

The most popular strategy was to maximize value or expected value and then ensure the plan met the time and attrition constraints. The second strategy was to maximize value while simultaneously constraining time and attrition.

Other search behavior included some users only changing one weight at a time between searches. In Level 2, some participants used the sensitivity analysis to make the smallest change necessary to alter the solution. This method was possible in Levels 0 and 1 only if the participant made incremental changes to a weight, clicked the search button, and repeated the

process until a new plan was generated. Most participants did not use this technique in Levels 0 and 1 and made large coefficient changes to generate new plans. Some participants also used knowledge from previous scenarios to guide their search. Most users changed only the time, value, and risk weights, but at least one participant used the utilization weight to change the plan. Some users described their search strategy as "trial and error." Other than the impact of the sensitivity analysis, there did not appear to be a difference in search techniques over the three experimental levels.

Reviewing the responses to various questions revealed that their evaluation of plan quality aligned with the two strategies described above. The participants' most important consideration was the expected value or value metric. The secondary consideration was ensuring that the time and attrition constraints were met or only slightly exceeded.

Two participants commented that in addition to the quantitative metrics, they also considered qualitative aspects of the solution when choosing a final plan. An example of a qualitative strategy applied by a participant is trying to prosecute targets near a specific location or in a straight line from the UAV base to a specific location. These qualitative attributes could not be captured with the current search algorithm, but they can be incorporated by adding mobilities to targets. Klau, et al. (2002) successfully added mobilities to delivery locations in their vehicle routing algorithm ("HuGS") that allowed users to incorporate qualitative plan metrics into their search strategy.

6.3.2 Decision Space Visualization

The Decision Space Visualization tool (DSV), discussed in Chapter 2, was developed by Draper Laboratory to compare solutions containing many metrics. The DSV was used to plot all of the HMCDM plans and the one global plan generated by each participant in each experimental level. The plans ranked in scenario 4 were also plotted separately. Because there are too many plots to include, only a few interesting ones will be discussed. All plots can be found in Appendix B. It is important to note that the users did not have access to the DSV and did not have a method of comparing all of the plans, except with the Metric History in Level 2. However, they did have access to the plan comparison window, which allows them to select up to four plans to compare from a chronological list of all the plans generated.

The vertical axis of the plot is either value or expected value, depending on what was inferred from the questionnaire as the user's preference in comparing plans. The horizontal axis is expected aircraft attrition. Colored markers are used to plot the plans. The color of the markers is dependent on the plan's time to complete the mission. A color scale is located beneath the chart. These metrics were selected because they were emphasized in the scenarios and the users commented that they used these metrics to evaluate plan quality. The default shape of the markers is a circle. The markers are square if they are on the local expected attrition/expected value or value Pareto frontier. A vertical yellow band marks the point on the graph where plans to the right of the line violate the constraint of losing more than one UAV. Plans with a number above them have comments in the left panel of the interface. An orange circle surrounds plans of interested that are discussed in this section. These highlighted plans *might* have been better than the user-selected HMCDM plan.

For each experimental level, arrows show what plan is the HMCDM or global plan. The plan with a bulls-eye around it indicates the plan selected by the user to be the best for that scenario. In the bottom left panel, the participant explains which plan he selected and why.

For each set of scenario 4 rankings, plans labeled with an 'H' are HMCDM plans and those with a 'G' are global plans. The number following the 'G' indicates the number of times the participant used the global search. For example, 'G1' indicates that the plan is the first time the participant ever used the global search. 'G3' indicates that the plan is the third and final time the participant used the global search. The purpose of the number is to determine if experience with the global search yielded improved global plans. The number above the plans reveals the ranking of that plan. Finally, the comment in the bottom left panel explains the participant's rationale for selecting the number one ranked plan.

The Participant 1 (P1) Level 0 plot, Figure 47, shows three distinct positive sloping bands of plans generated with the HGA search. The bands reveal an ordered search strategy that progresses from low to high valued plans. A three dimensional plot of this graph can be seen in the 3D Plot section. The global plan fits nicely into the top band, but violates the attrition constraint. P1 selected the highest valued plan that met the attrition constraint. The HGA steering and decision made by P1 in Level 0 appear very logical.

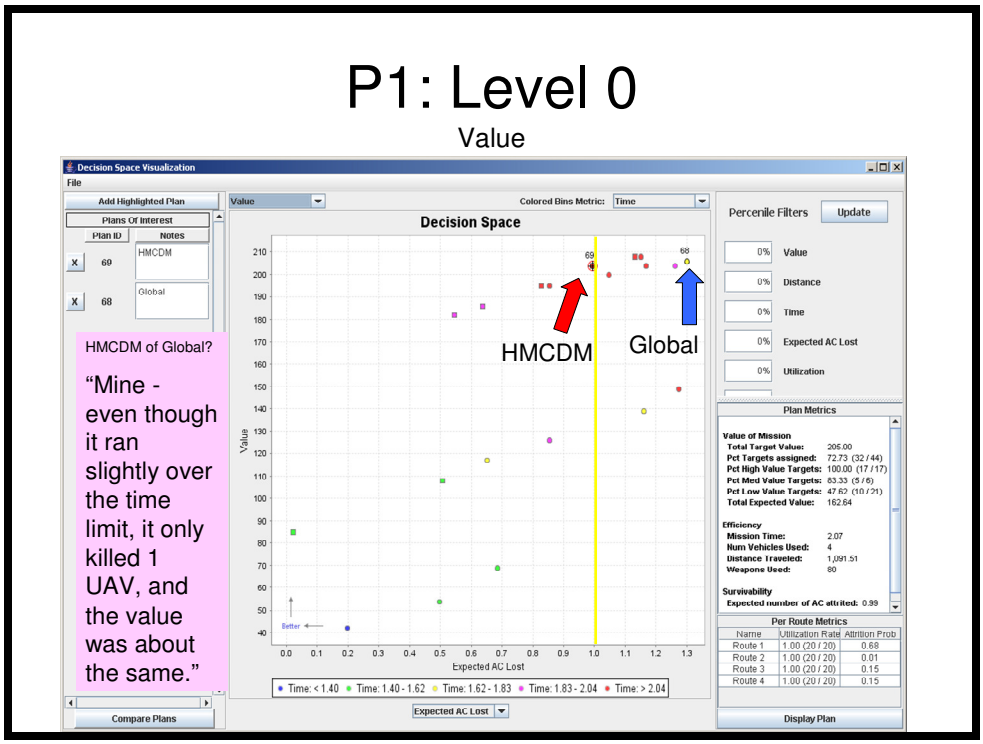


Figure 47: DSV Plot - Participant 1 Level 0

A study of Participant 2's (P2) plots shows a trend seen in most of the other participants. The Level 0 plot, Figure 48, highlight two plans with orange circles that are superior to the global and HMCDM plan with respect to expected value and attrition. The far left plan meets the time constraint of 2 hours. If P2 had access to the DSV, he *might* have selected one of these highlighted plans, especially the one on the far left as the superior plan. A similar result can be seen in P2's Level 1 plot, Figure 49.

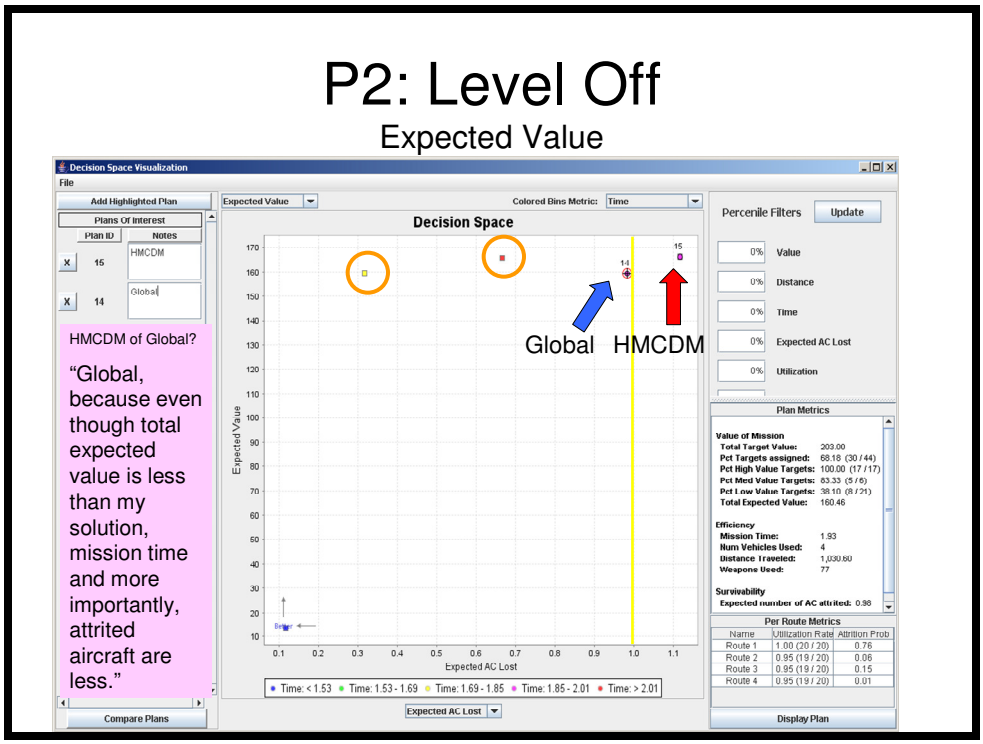


Figure 48: DSV Plot - Participant 2 Level 0

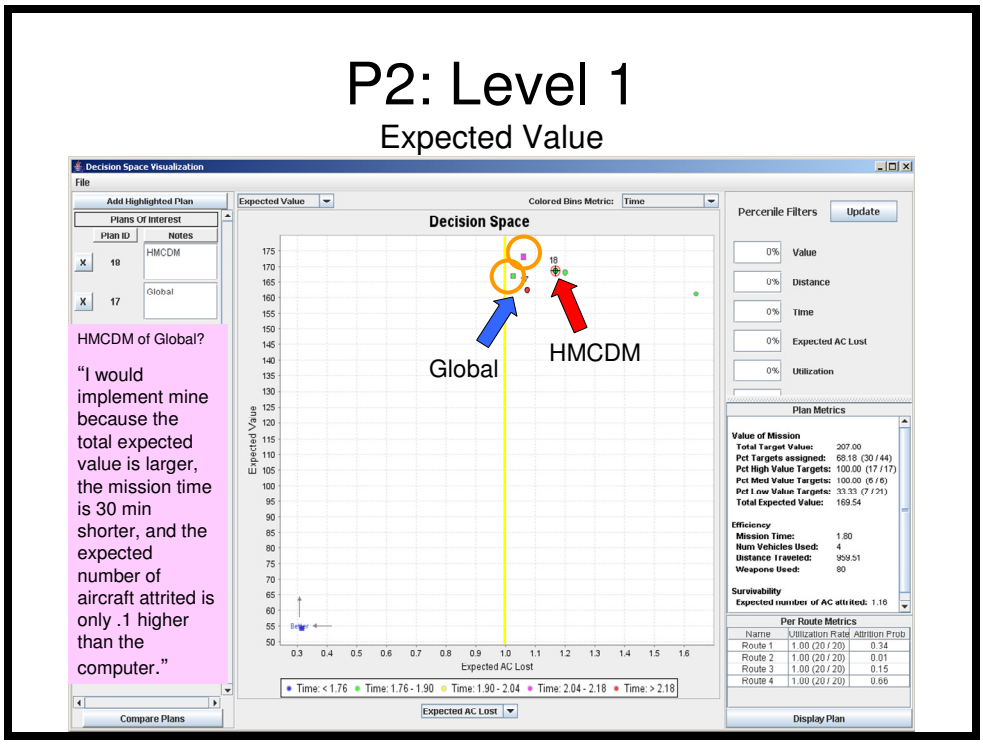


Figure 49: DSV Plot - Participant 2 Level 1

Plotting the highlighted plans from Levels 0 and 1 on P2's scenario 4 rankings, Figure 50, shows that some of these plans *might* have changed his rankings.

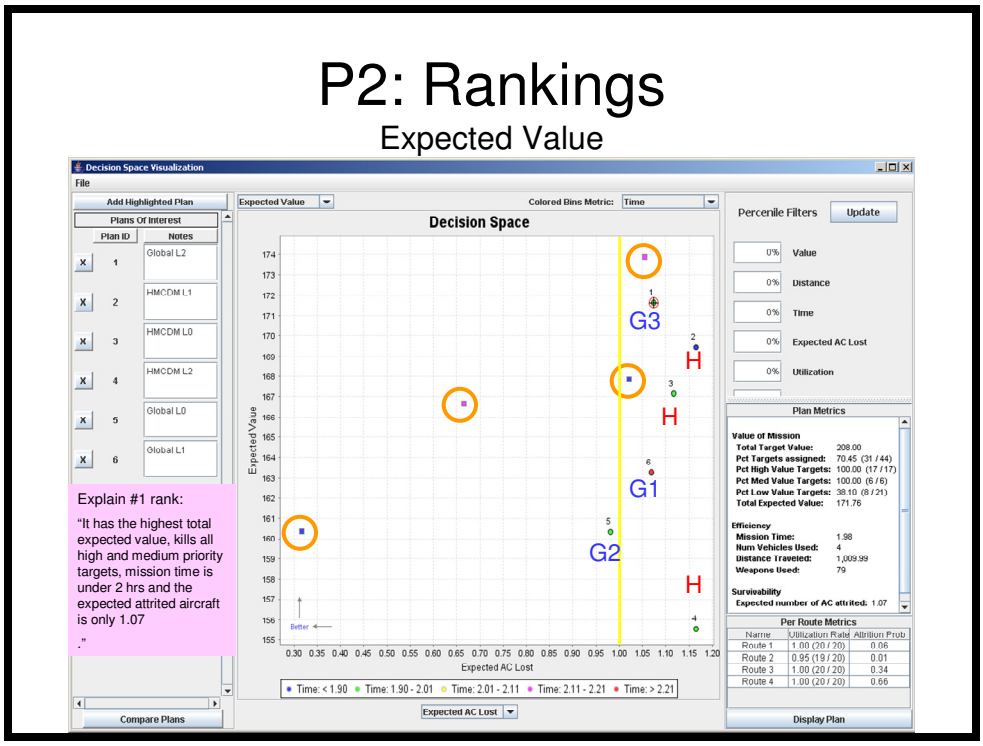


Figure 50: DSV Plot - Participant 2 Scenario 4 Rankings

When asked which search method, HGA or global, the participants would use if they needed to complete another scenario, P2 is the only one who preferred the global search to the HGA. It may be possible that having access to the DSV, or some other method of comparing and selecting HGA plans, his decision would be different. Since this trend is seen in most participants, a quality method of decision space visualization would likely benefit most users when they need to compare plans. It must be cautioned that this analysis is very subjective and it cannot be known exactly how the participants evaluated plan quality. Therefore, no conclusions can be drawn regarding whether having access to the DSV would affect the experimental outcomes. P1's rankings, Figure 51, show his tolerance for risk and what tradeoffs he made in his decision making. The highest ranked plan is the plan with the highest value that meets the expected attrition constraint. The second ranked plan violates the attrition constraint with an attrition value of 1.07 UAVs. The third ranked plan has a lowest value of all the plans, but does not violate the attrition constraint. The fourth, fifth, and sixth ranked plans all violate the

constraint with increasing attrition rates of respectively 1.17, 1.29, and 1.62 UAVs. In selecting which plan to rank third, P1 likely decided that the fourth ranked plan's 1.17 UAV attrition was not worth its 13 additional value points compared to the third ranked plan. This is information that the computer could intelligently capture and use to narrow the decision space, highlight plans for future consideration, and possibly suggest weights for future HGA searches.

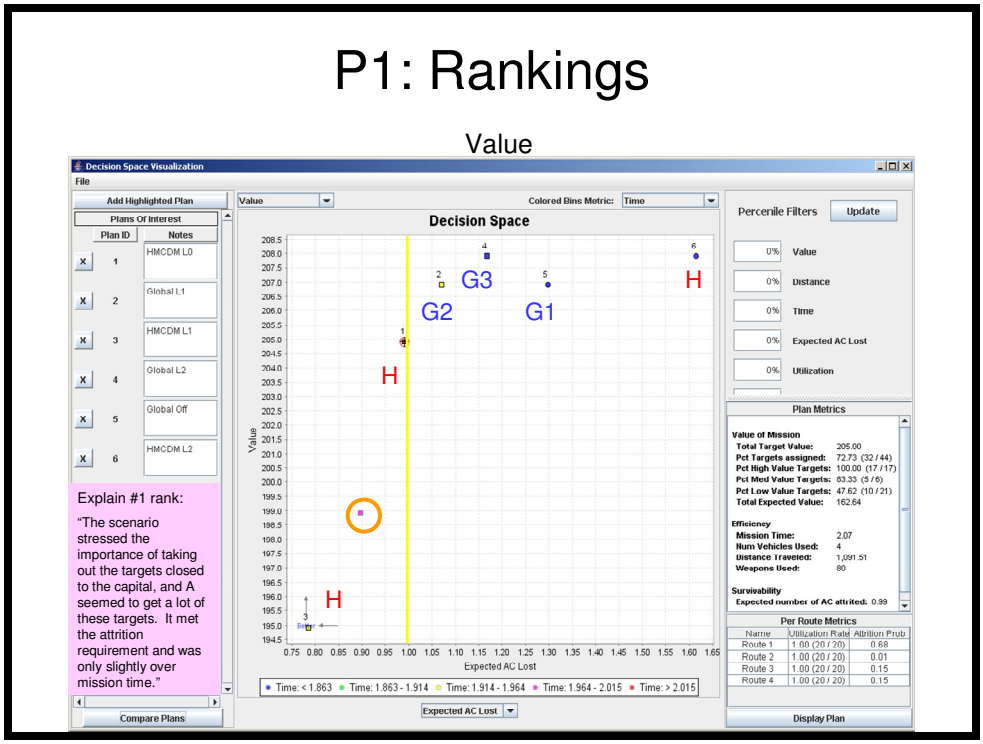


Figure 51: DSV Plot - Participant 1 Scenario 4 Rankings

From the DSV plots, there is not enough evidence to conclude whether the Metric History had an effect on the participant's abilities to compare the plans in Level 2. Most of the highlighted plans appear in Levels 0 and 1, but that is likely because more plans were generated in those levels than in Level 2. DSV plots show that the first global search consistently had low rankings as discussed in the HGA Effectiveness section. However, there did not appear to be any additional patterns between the user's experience with the global search, and the quality of the global plans.

6.3.3 3D Plots

Three-dimensional plots were graphed in MATLAB to visualize the solution space (in three of the five dimensions) and compare the global and HGA searches. These plots were not available to the participants during the experiment. All 3D plots can be found in Appendix C. A few interesting plots are discussed in this section.

The axes were determined by the three metrics emphasized in each scenario: expected value, expected aircraft attrition, and time. The z-axis is expected value, and the x and y-axes are either time or expected attrition, depending on the graph's rotation. Each graph is rotated to a position that allows the best view of the plot. The solid red circles represent the plans generated by the HGA and they are connected with a red line to show the order in which they were generated. The large solid red circle is the HMCDM plan. During the experiment, only the search coefficients and the global plan were saved. Using these coefficients, the global search was run again after the experiment for 15 minutes and all of the plans were saved. The plans considered by the global algorithm in the post-experiment search are represented by black dots; these are the plans visited by the global search and do not include the neighborhoods of these plans that were also considered. The solid blue circles represent the best global plans as the algorithm searched the solution space during the post-experiment search. The blue line is the global plan search path, which shows the order that the best global plans were generated. The large solid blue circle represents the global plan. Since the global plan is the best global plan found during the experiment and the global plans on the plot were found after the experiment, the global plan might not be on the blue line. Finally, the participant-determined best plan for that scenario, either the global or HMCDM plan, is circled in green.

P1's Level 0 graph, Figure 52, shows his search spiraling up to a high expected value, resulting in the three distinct bands found in P1's DSV plot. This spiral is not seen on P1's Level 1 or 2 plots (see appendices). Level 0 was P1's first scenario, which can help explain the spiraling. Because a plan that does not send any UAVs to destroy targets was not a viable solution, the solution that used the least resources (UAVs, time, attrition, fuel) was a plan that sent one UAV to prosecute the closest target to the UAV's home base. As explained in Chapter 5, this plan is called the Null plan. The Null plan was the starting plan for the HGA search at the beginning of each scenario. For the user to escape the Null plan's neighborhood of low valued plans, where only one or two UAVs may be utilized, the user must set the value coefficient to be

relatively higher than the other coefficients. This method would ensure that the next plan generated would utilize all four UAVs and generate high-valued plans. This technique was emphasized and practiced in the experiment's training. As P1 began his first scenario, he spiraled away from the Null plan and into an area of the solution space that better met his objectives. As seen in the Level 1 and 2 plots (see appendices), P1 learned from the first scenario and could immediately hone in on a high-valued area of solution space in the remaining scenarios.

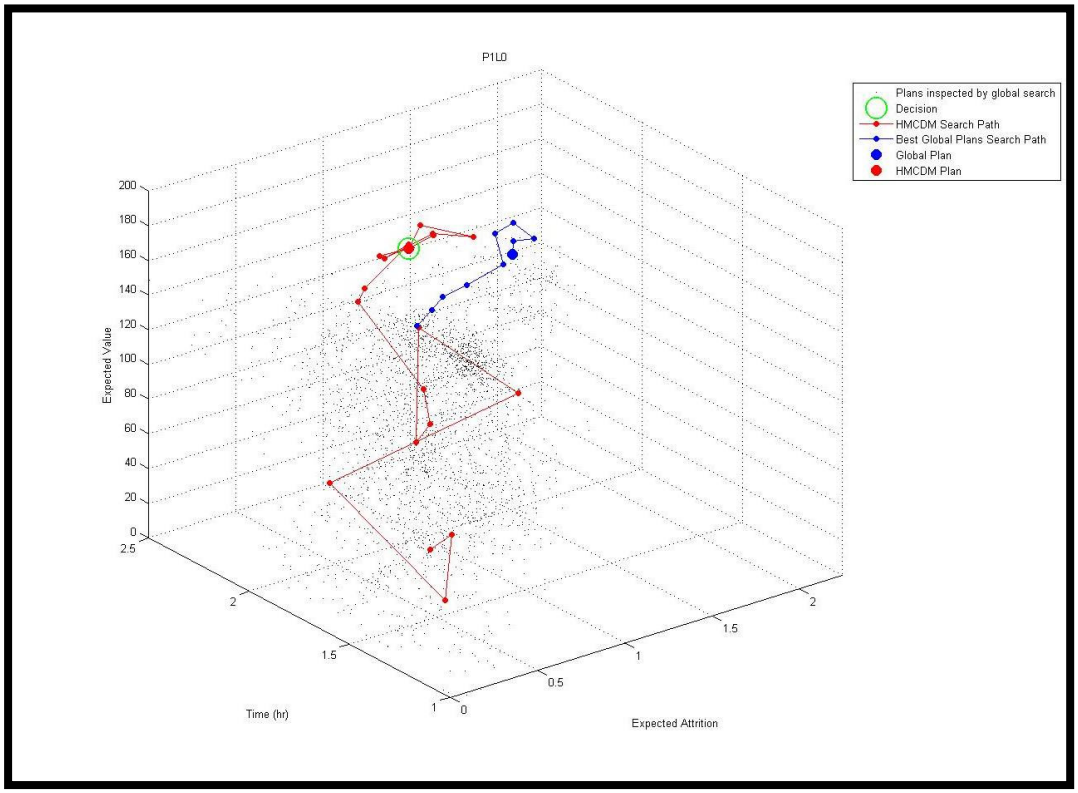


Figure 52: 3D Plot - Participant 1 Level 0
Notice the spiral away from the Null plan and into a high expected value area of the solution space.

P1's Level 0 plot also highlights the thousands of plans considered by the global search that were required to produce the global plan search path and the global plan. P1 used only 19 plans to hone in on the HMCDM plan, which he determined to be better than the global plan.

The exploratory search strategy used by Participant 6 (P6) is apparent from his 3D plots. P6 would consistently return to the Null plan during the search, and then try new coefficients in order to explore a new area of the solution space. P6's strategy can be seen in all levels of the experiment (Level 0 shown in Figure 53). Participant 4 (P4) and 7 (P7) also used this

exploratory strategy in most of the experimental levels (P4 Level 1 shown in Figure 54). In contrast, participant 5 (P5) and 8 (P8) kept their searches clustered in a tight area of the solution space (P8 Level 0 shown in Figure 55). These search strategies are different from the ones described by the participants on the questionnaire.

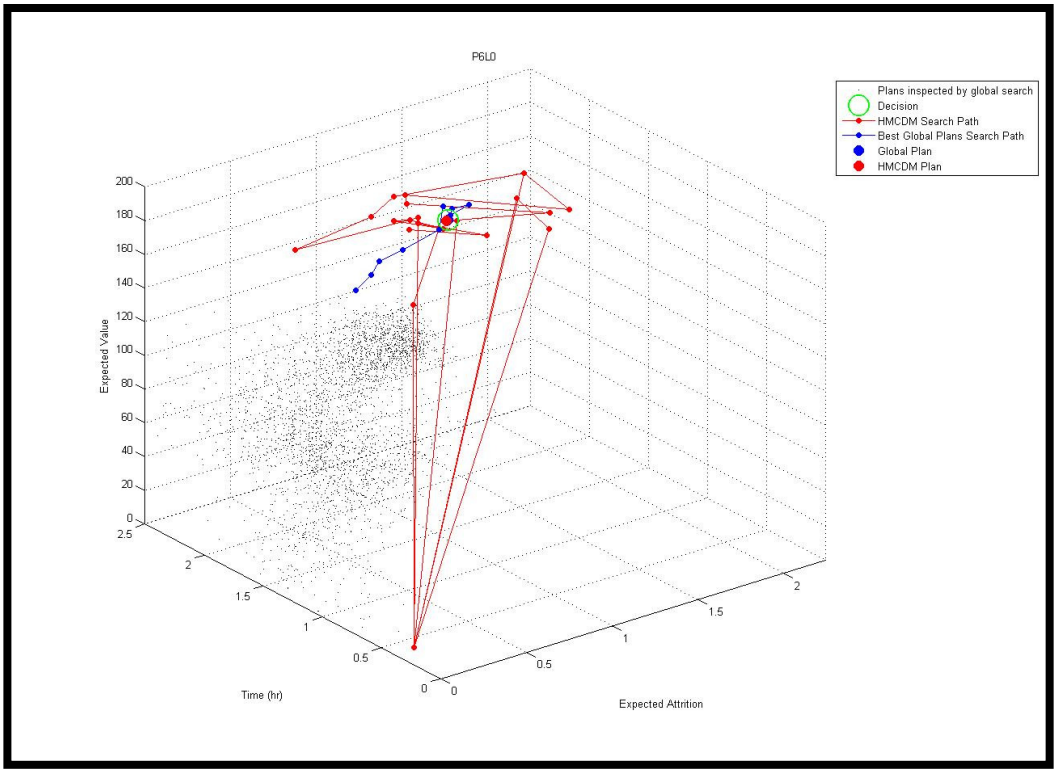


Figure 53: 3D Plot - Participant 6 Level 0
Notice the consistent return to the Null plan strategy.

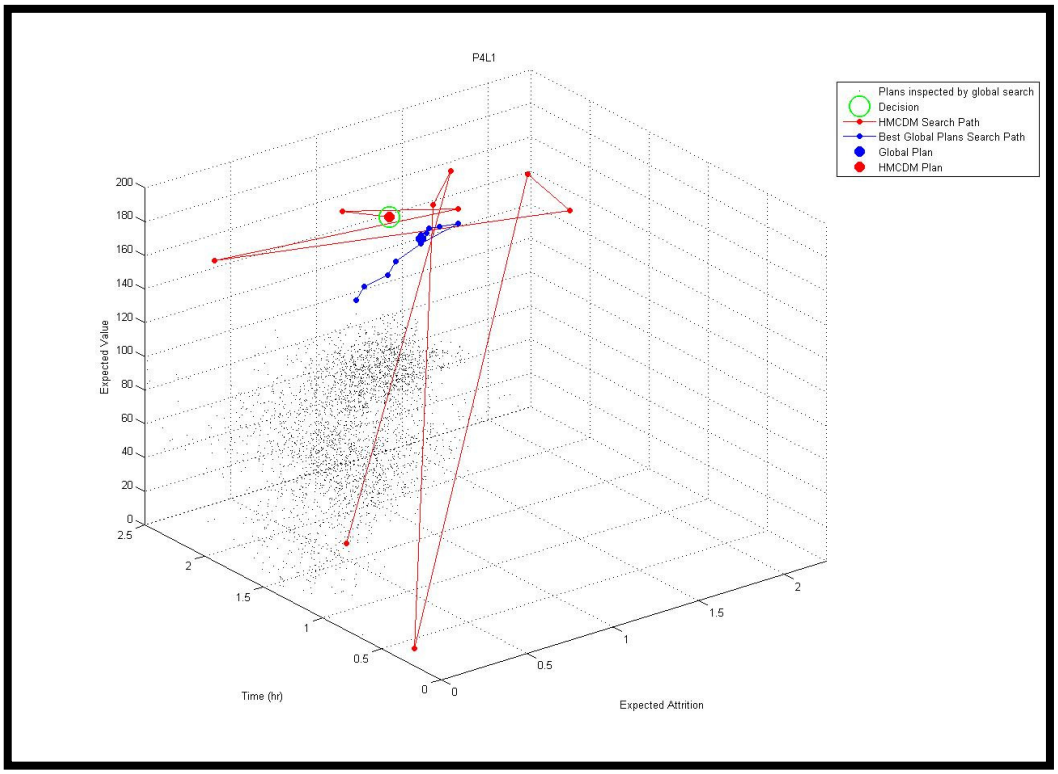


Figure 54: 3D Plot - Participant 4 Level 1
Notice the exploratory search strategy.

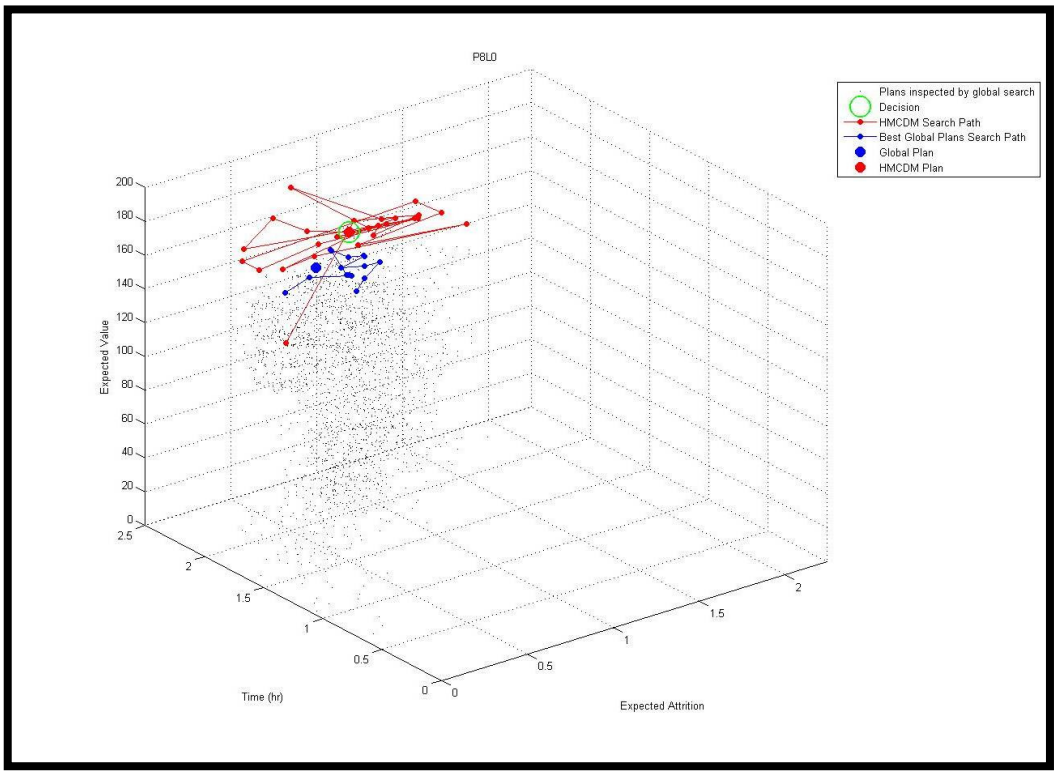


Figure 55: 3D Plot - Participant 8 Level 0
Notice the clustered HGA search strategy.

The fact that fewer unique plans were generated with increasing levels of TBD, resulted in some cases in a smaller area of the solution space being explored than in the other levels. There are not any other obvious differences in the 3D plots across the experimental levels. Because the participants did not have access to the 3D plots, they might not have known if their search strategy was clustered or exploratory. When asked about search strategy during the experiment, no comments mentioned these search techniques. The participants' strategies used across the levels also appeared to be consistent. The 3D plots do not reveal the strategy effect of the Sensitivity Analysis Display.

The 3D plots highlight the strengths and weaknesses of humans and computers. Humans could consistently steer the HGA and hone in on an acceptable area of the solution space. However, within the 15-minute time period, the human could only generate tens of unique plans. Within 15 minutes, the computer could search thousands of unique plans, but there was no guarantee that the global plan would be an acceptable solution. As seen in some of the above plots, the global search focused its search in a clustered area of the solution space while the human had the ability to search a greater area. The Future Work section of Chapter 7 offers suggestions on how these human-computer complimentary strengths can be applied to solve better the UAV routing problem.

6.3.4 Coefficient Plots

MATLAB was used to plot the weights used for each HGA search. Each plot contains four subplots that can be used to compare how the user selected coefficients across experimental levels. The x-axis is the number of searches performed. The y-axis is the weights. The first subplot shows the weights of the Level 0 search and its y-axis is scaled from 0 to 100, like the Sliding Bars. The second subplot standardizes the weights of the Level 0 search so that they sum to 1. This allows for a comparison between the actual weights on the first subplot and their relative values in subplot 2, as would be shown on a pie chart. The third and fourth subplots respectively show the standardized weights for Levels 1 and 2, as would be shown on the pie chart. The legend explains which colors and shapes represent each weight. All of the Coefficient Plots can be seen in Appendix D**.**

Looking at P7's plots, Figure 56, the Level 0 standardized weights subplot (subplot 2) is significantly different from the Level 0 actual weights subplot (subplot 1). The sliding bar coefficient for value was kept at 33 for 35 of the 36 searches. Because the value sliding bar was not moved during the searches, the user might infer that the value weight remained constant. However, the standardized plot shows how the value coefficient varied considerably during those searches, from .33 at the first search to .91 at search 34. These subplots highlight the reason the Sliding Bars can be a deceptive means of controlling the HGA. Other than the number of searches performed, there are not any obvious differences in the coefficient plots across the experimental levels.

P6's and P4's coefficient plots show how they changed the weights to frequently return to the Null plan and explore the solution space (P6 shown below in Figure 57). Both P6 and P7 were identified earlier as using an exploratory strategy; however, their coefficient plots look very different (both shown below). The coefficient plots are not as informative as the DSV or 3D plots. Because the coefficient plots are identical to the Weight History in Level 2, the Weight History alone would probably not be useful to the user, unless directly compared with the Metric History or another method of visualizing the solution space.

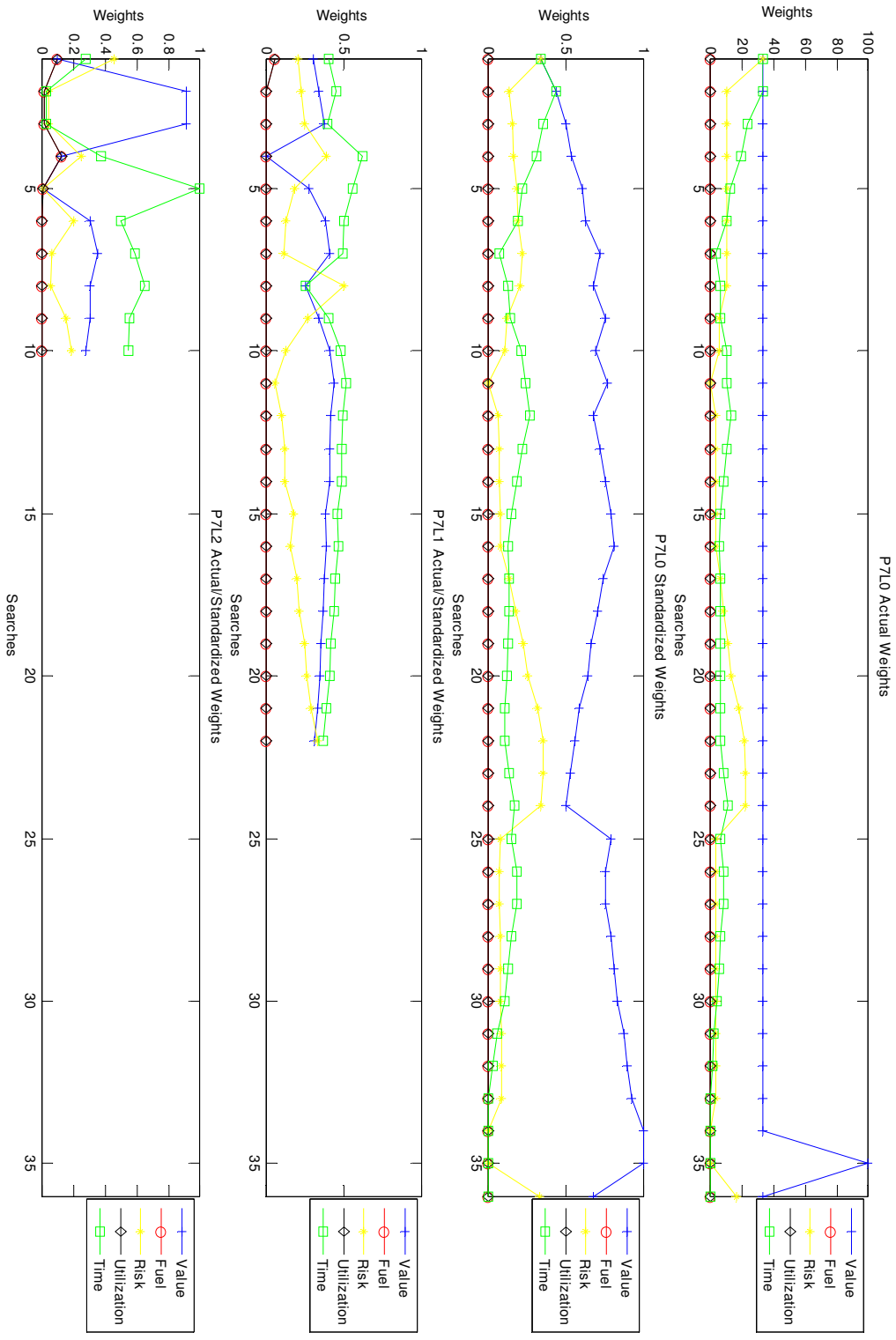


Figure 56: Coefficient Plot - Participant 7

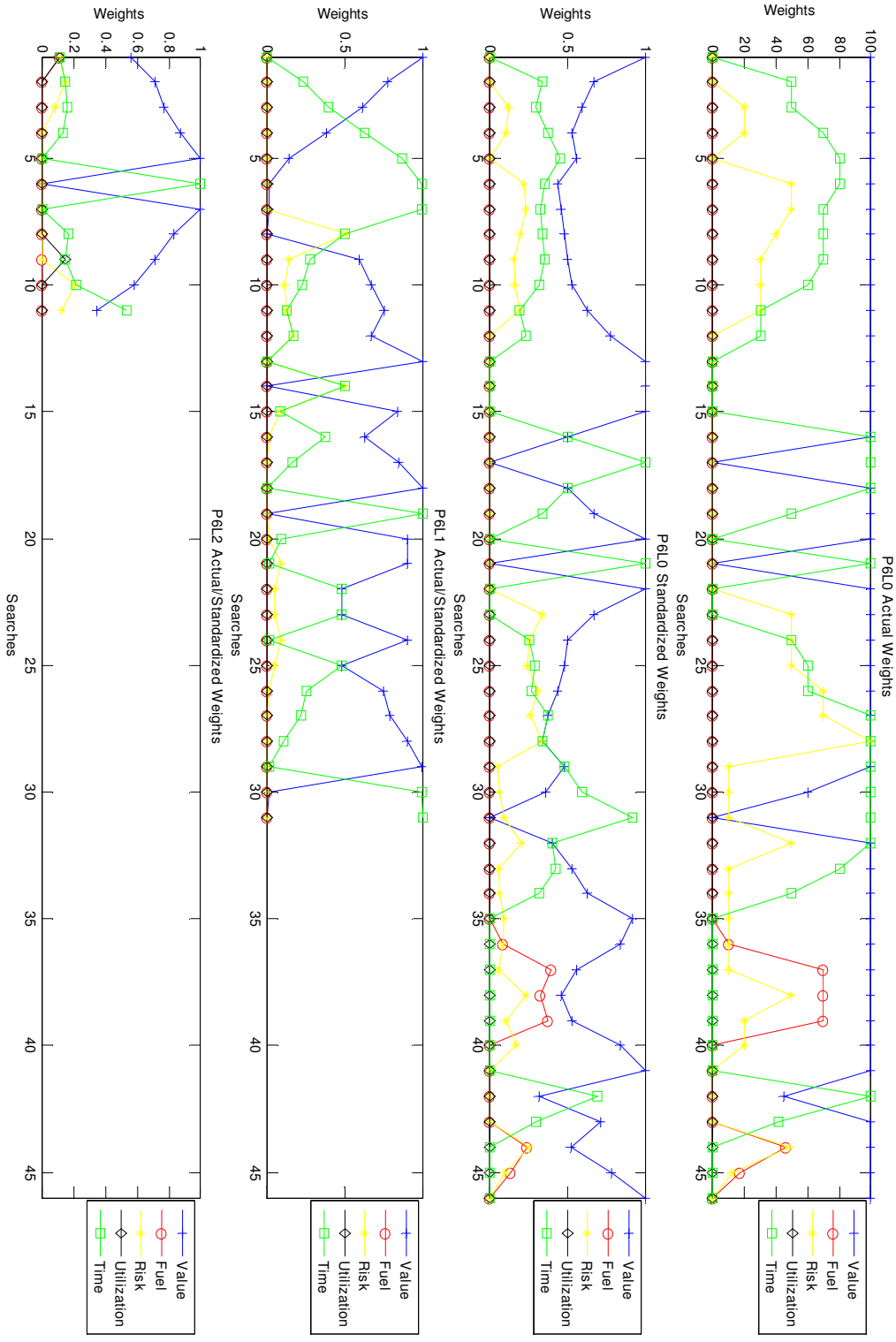


Figure 57: Coefficient Plot - Participant 6

6.4 Discussion

TBD had a significant effect on the participants. Increased levels of TBD improved the users' understanding of the algorithm, ability to predict the affect of changing the weights, ability to steer the HGA, and increased faith and trust in the HGA, and confidence in the HMCDM plan. While some of these results are approaching significance, a larger sample size might have made them significant. It is interesting to note that in violation of Rempel's (Muir 1994) theory of trust development, TBD developed faith in the HGA before dependability. Because the user had relatively little experience with the HGA over the entire experiment, the observed measure of faith might have been "blind" faith, which is not a good attitude to develop. TBD accomplished the above improvements without a significant change in operating difficulty. These results were likely due to improved design of the controls, increased transparency, and additional information provided to the users to evaluate and understand HGA performance. TBD in this experiment improved the participants' trust in themselves and their ability to operate the HGA. This resulted in improved confidence in the HMCDM plan.

It is necessary to discuss the implications of blind faith and increasing trust in the higher-level TBD HMCDM plans despite the fact that these plans were not superior to lesser-trusted plans. The increase in trust without an improvement in plan quality can be explained by the user's involvement in generating plans. Using the HGA to generate plans, provided the user a comparison of the HMCDM plan to all the other plans generated in the search. Because increasing levels of TBD improved methods of comparing plans and searching the solution space, it is theorized that TBD improved the basis of comparison for evaluating plan quality, thus leading to increased trust in TBD methods of plan generation. There are two ways to evaluate this effect. First, because the purpose of TBD is to calibrate appropriate trust in the HGA and the HMCDM plan, improving trust without an improvement in plan quality can have serious negative consequences if the trust in the plan is too high. Second, it is possible that plans not generated with TBD tools were under-trusted and TBD raised trust to appropriate levels. In this experiment, it was not possible to determine what interpretation is correct. This shows a weakness of the TBD tools in this experiment. The tools do not reveal how the plans will actually perform in real-world implementation; they are designed only to allow users to evaluate, understand, and improve performance of the HGA. A simulation was not available to evaluate plan quality across all participants and compare user trust in the plan to the plan's actual

performance in simulated real-world scenarios. To prevent dangerous situations of inappropriate trust, trust information should be used to supplement TBD tools. A simulation can be used to provide trust information for a more accurate depiction of how to trust the final plan. In addition, the possible development of blind faith in the experiment might have been prevented with trust information that provides historic data of HGA performance.

The survey results confirm the need for trust information. Most participants believed that trust information would affect their decision making. Despite being informed of variability in the plan metrics, only one participant built in slack time to meet the constraints of the scenario. This indicates that users need help knowing what information to trust, when to trust it, and how to trust it. It is theorized, that HGA users can successfully incorporate trust information into their decision making to generate robust plans. Providing the users with information about the accuracy of the data, metric calculations, models, and heuristics that many users blindly rely on allows the users to make better-informed decisions and compensate for weaknesses and variability. Trust information also gives the users a measure of reliability in their final plan, therefore improving trust calibration. Trust information supplies users with a better understanding of the uncertainty in their environment and the vulnerability of their plan. It is theorized that with trust information, the users can steer HGAs to robust solutions that are therefore more trustworthy than those created without this information. A UAV routing simulator is necessary to test this hypothesis by comparing the simulated performance of plans generated with and without trust information. For users to develop robust plans, resolution and specificity of trust information is necessary.

With respect to trust specificity and resolution, the participants' comments indicate that they could separate the trust implications of the entire HGA and the individual components, most notably the HGA control. This result is not surprising, because the most noticeable difference between each level of the experiment was the control of the HGA. Most participants commented about the HGA controls, rather than the displays, revealing the significant impact the control had on them throughout the experiment. The most commented item was the positive attributes of the sensitivity analysis.

TBD of the control used to steer the algorithm significantly improved the users' search strategy and understanding of algorithm logic. The Pie Chart Control with Sensitivity Analysis Display had the most noticeable effect on the participants. The control did not affect the users'

decision making. There was not a significant difference in the three different controls' difficulty of use. The Sliding Bars in one case gave a participant the false impression that changing one weight had no effect on the other weights. The Pie Chart Control correctly shows how changing one coefficient affects the others. Because the Pie Chart Control was not more difficult to use, and correctly displays the relationship between the HGA's objective function coefficients, it should be used to steer the HGA. Other advantages of the Pie Chart Control include improved controllability with the ability to fix coefficients to a constant value and the ability to type in values for the coefficients. For future improvements to the Pie Chart Control, including adding the capability for it to be manipulated with the mouse, see the Future Work section of Chapter 7.

The Sensitivity Analysis Display had the greatest effect on the participants and most of the significant results can likely be attributed to it. Most users valued the extra information provided by the Sensitivity Analysis Display, which allowed them to steer the algorithm more efficiently. Eighty-two percent of the searches with sensitivity analysis generated unique plans, while only 39% of searches without sensitivity analysis generated unique plans. However, the time required to calculate the sensitivity analysis significantly reduced the total number of plans generated in each scenario. Seven of the 8 participants selected the Sensitivity Analysis Display for future searches, concluding that the value of the sensitivity information was worth the computational time. In the future, for the Sensitivity Analysis Display to be effective in generating higher quality plans, it must be calculated quickly. This keeps the users focused and engaged on their task, rather than watching and waiting for the results to appear on the screen. See the Future Work section of Chapter 7 for ideas on how the sensitivity analysis can be improved.

The second most effective TBD display was the Rationale Window. One of the goals of the Rationale Window was to link the changes in the weights with the changes in the plan. Few participants commented about this aspect of the Rationale Window. It may be possible in the future to make this connection more apparent. Overall, the Rationale Window is a valuable tool for the HGA.

The Plan Cycle Option was not useful to the participants. It was interesting to watch the algorithm search the solution space, but it had little effect on decision making or search strategy. A few users commented that it helped to improve their understanding of the algorithm. Because it is important for the user to have an accurate mental model of the internal workings of the

HGA, the Plan Cycle Option might be useful in training the users by providing an example of how the local neighborhood search operates. The Plan Cycle Option is not a necessary or effective display for operating the HGA, but it should be an option available to interested users so they can build an accurate mental model of the algorithm.

The Weight and Metric Histories had mixed reviews from the participants. As commented by one of the participants, it takes time and experience with the HGA to recognize the benefits of the Weight and Metric History. However, the coefficient plots in the Operator Strategies section show that the Weight History might not be a useful display by itself. However, combined with the Metric History, both history graphs might be more useful. The DSV and 3D MATLAB plots display the solution space differently than the Metric History plot. The strengths of the Metric History include the ability to link the Metric History with the Weight History and the ability to quantitatively compare plan quality as the HGA search progresses. The results of the qualitative DSV analysis show that users should have a quality means of comparing their plans. Compared to the Metric History, the DSV is much better at comparing plans of multiple dimensions. The DSV shows plans along the local Pareto frontier, and allows users to change the axes and color scale, and to filter plans. Plotting plans generated by the HGA in the DSV likely improves user decision making by improving the users' ability to compare the plans. The 3D plot allows the users to visualize their search in three dimensions of the solution space, but the angle that the plot is viewed significantly impacts what can be seen on the graph. It is an interesting plot to analyze, but its only strength is its qualitative display of the HGA search path. This option may be useful for operators to see what area of the solution space they have visited and find new areas that they can explore. It may also help users develop new search strategies, as discussed below. These three methods of visualizing the solution space both have their strengths and weaknesses, but the DSV would likely be the best display for the HGA. For ideas on how to improve the solution space visualization and combine the strengths of the above displays, see the Future Work section of Chapter 7.

The 3D plots revealed the strategies used by the participants. The combined effect of the Metric and Weight History had little effect on strategy. The participants used different strategies, which is a result seen in previous experiments discussed in Chapter 2. Because the plans across the participants cannot be compared, it is not possible to evaluate the effectiveness of the search strategies employed in the experiment. The 3D plot of the solution space might

allow users to ensure they visit unexplored areas of the solution space. By trying different search strategies, participants can determine what methods are best for solution space exploration. The 3D plots might be the most promising display to help users develop and fine-tune their search strategies.

With 7 of 8 participants selecting the HGA as their preferred method of generating plans, when also given the option of the global search, the utility of HGAs is confirmed. As shown by the 3D plots, humans can steer the HGA to solutions by looking at far fewer plans than required by the global search. Humans cannot evaluate plans as quickly as the computer, but humans are superior at incorporating human or qualitative plan objectives and then steering the algorithm to improve the plan. Depending on the problem being solved, the algorithms available, and the development of new search algorithms, heuristics, and collaborative techniques, the role of the HGA must adapt to best match the strengths of both computers and humans. In this experiment, the HGA was superior to the global search, but that might not be true for other applications. For ideas on how to improve the human-computer collaboration of the HGA used in this experiment, see the Future Work section of Chapter 7. For ideas on how HGAs can be applied to other areas of current research, see the Future Work section in Chapter 7.

Despite the aforementioned improvements made to the HGA by applying TBD, the plans generated in the different levels of TBD were not significantly different in quality. As predicted, TBD significantly decreased the number of searches performed. Providing the user with additional information with TBD tools slowed the participants' ability to rapidly generate plans. It was also slower to set the weights with the Pie Chart Control than with the Sliding Bars. With respect to the number of plans generated, the participants were able to generate plans of the same quality as the control (Level 0) with fewer searches in Levels 1 and 2. However, with respect to the 15-minute search time, the plan quality was not significantly different across the levels of TBD. Since HGAs are likely to be beneficial in time-critical situations, improving plan quality with respect to time is an important consideration of HGA trustworthiness. Improving plan quality with respect to time was not accomplished in this experiment, which likely resulted in the lack of improvement in plan quality with increasing levels of TBD. See the Future Work section of Chapter 7 for ideas on how high quality plans can be generated faster.

Overall, TBD was a success, but more research needs to be done. According to scenario 4 rankings, Level 1 plans were almost significantly better than Level 0 plans. This can be

attributed to the benefits of TBD. However, Level 2 plans were almost significantly worse than Level 1 plans. This can be attributed to a side effect of the sensitivity analysis, the most successful TBD tool. It is theorized that if the time required to calculate the sensitivity was reduced, a combination of two possibilities will occur. One possibility is that Level 2 will produce more unique plans than Levels 0 and 1 in the same amount of time. With more plans to choose from, Level 2 will likely produce plans superior to Levels 0 and 1. The second possibility is that Level 2 will quickly hone in on plans equal in quality to those that can be produced in Levels 0 and 1, making it possible to generate equivalent solutions in less time. Of course, the third possibility is that none of the above occurs. This Sensitivity Analysis Display example shows why it is necessary to strike a balance between the costs and benefits of TBD so that TBD can both improve HGA effectiveness and foster appropriate trust calibration.

Chapter 7

Summary and Future Work

This thesis has merged trust research with Human Machine Collaborative Decision Making (HMCDM) to solve complex problems by developing solutions that can be appropriately trusted. This chapter summarizes the work presented in the thesis and suggests ideas for future research.

7.1 Summary

Current Operations Research approaches model and solve complex problems with the aid of powerful computers. By combining the strengths of human and computers, HMCDM has been shown to generate higher quality solutions in less time than traditional methods (Malasky 2005, Forest, et al. 2007). In some cases, it is difficult to model constantly changing problems and incorporate human objectives into the solution. Human-guided algorithms (HGAs) harness the power of sophisticated algorithms and computers, but provide flexibility to the human decision maker to model the problem correctly and steer the algorithm to solutions that match his objectives. HGAs are designed to make the power of Operations Research accessible to problem domain experts and decision makers.

Like any type of automation, HGA operators must appropriately trust the HGA and the final solution. Through the use of trust-based design (TBD), it was theorized and tested whether HGA users will gain insight into the solution process, improve their calibration of trust, and generate superior solutions. Participants operated an HGA that routed military aircraft to prosecute targets for the purpose of testing the effectiveness of HGA controls and displays that incorporated TBD. It was shown that TBD had a significant effect on trust, HGA performance, and in some cases the quality of final solutions. It was also confirmed that for proper trust calibration, HGA operators must be provided with trust information to improve their understanding of the HGA, the solution process, and the final solution.

7.2 Future Work

These suggestions are made for researchers and Draper fellows interested in continuing research into HMCDM, HGAs, and TBD.

7.2.1 Applications to Current Research

Various researchers at the Draper Laboratory have expressed interest in incorporating TBD into their projects. It must be emphasized that TBD is not an add-on to current applications and algorithms, but should be a consideration at the start of the design process. TBD does not solely develop computer interfaces, but reaches much deeper into the structure of the algorithms and how the human and computer can work together to solve problems. It then provides controls, displays, and information that maximizes the benefits of human-machine collaboration and can generate superior solutions in less time than traditional solution techniques.

TBD tools can be used to select coefficients and parameters used to track satellites (Johnston 2007) and to plan schedules robustly for military transport aircraft (Martin 2007). An HGA can be developed from an algorithm that safely routes military troop convoys in dangerous environments (DeGregory 2007). Involving commanders in the HMCDM process will likely improve appropriate trust in the plan and the HGA. A military convoy routing TBD HGA could make Operations Research accessible to troops on the battlefield and result in the saving of lives.

7.2.2 Improvements to UAV Routing HGA

The Sensitivity Analysis Display had the greatest impact in the experiment, but its computational time must be reduced. Because most users focused on changing one coefficient at a time, an option that allows the user to select which sensitivities to calculate reduces the time required for the calculation. The average sensitivity calculation lasted approximately 10-30 seconds. If the user selected one or two weights that require sensitivity analysis before each search, the time waiting for the sensitivity could be brought within a reasonable amount of time (5-15 seconds) and allow for a greater number of unique plans to be generated. This option gives more control to the users by allowing them to decide when and what type of information they need, and therefore better manage the computational effort of the computer. Of course, this improved

sensitivity analysis control ability must be balanced against the effects of overwhelming the user with increased interface complexity and responsibility for guiding the HGA.

Five of 8 experiment participants said they would like to adjust the Pie Chart Control with the mouse. While a few users commented that using the mouse would not be necessary, others said that it would be an interesting alternative and force them to look at the pie chart graphic more often. Giving the users a choice of both typing weights and using the mouse likely has no negative consequences and provides interested operators with an easier method of steering the algorithm. Using the mouse might be a faster alternative than typing numbers, as the mouse operated Sliding Bars were the fastest method of changing the weights.

As discussed in Chapter 6, the Metric and Weight Histories, Decision Space Visualization (DSV), and 3D plots of the plan metrics had their own positive attributes. Linking the DSV to the HGA allows the operator to better compare plans against one another when selecting the final solution. Allowing the operator access to all three visualizations might be helpful during different phases of the search. Ideally, all the information should be presented in one visualization. Two display aspects that can be improved are 1) the link between how the weights affect the metrics and 2) the localness of each search. The most difficult concept for the participants to grasp was that each HGA search was local, and that exploration of the solution space was necessary to globalize the entire solution generation process.

The global search algorithm can consider many more plans than the human can in the same amount of time. It could be possible for the HGA to implement a global search along with the local search. One option is for the operator to steer the global search dynamically with the Pie Chart Control. Another option is for a dynamically steered global search to be used in conjunction with the local searches. The global search could offer its best-known solution for the user to consider. The user could then perform local searches from that solution or make manual modifications to it. The strengths of the global search algorithm could add value to the current HGA.

Finally, the use of mobilities, as used by MERL (Klau, et al. 2002, “HuGS”) and explained in Chapter 2, could add a tremendous amount of flexibility and controllability to the HGA.

7.2.3 Further Development of TBD and Trust Information

Additional research is needed to evaluate how trust information will affect the use of HGAs and decision making. Instead of approaching a new decision tool with the attitude that the best information and solution techniques lead to the best possible solution, a more responsible approach is to understand the uncertainty and vulnerability in the decision process and the final solution. The solution might be the best possible, but it also should be trusted appropriately. Trust calibration must be considered by designers of decision-making tools if they want their products to be used correctly. However, incorporating humans in the decision-making processes confound traditional methods of risk assessment and probability analysis due to the unpredictability of humans. Providing an appropriate level of trust information to an HGA operator remains a challenge for future research. This experiment determined that decision makers are interested in this type of information and that it would likely affect their decision making.

7.2.4 Additional HGA Research

With appropriate trust information, it is theorized that humans can generate robust plans. Because human problem domain experts are skilled at understanding where a plan will fail, their knowledge should be incorporated into every solution. Allowing these experts to steer HGAs to solutions that make the plans less vulnerable should make them more durable in complex, uncertain, and constantly changing environments.

Another area of interest is investigating the human strategies used to operate an HGA. Determining what strategies are effective, why some humans are better in certain situations than others, how teams of users can operate an HGA, and other insights into how humans interact with HGAs will aid the development of future HGAs.

Appendix A

Experiment Questionnaire

- How would you rate the predictability of how the weights you input would change the current plan?

1	2	3	4	5	6	7
Very Unpredictable	Unpredictable	Fairly Unpredictable	Neutral	Fairly Predictable	Predictable	Very Predictable

- To what extent can you depend on the HGA testbed to aid you in finding a good enough or optimal solution?

1	2	3	4	5	6	7
Very Undependable	Undependable	Fairly Undependable	Neutral	Fairly Dependable	Dependable	Very Dependable

- What degree of faith do you have that the HGA testbed will be able to cope with a large variety of scenarios?

1	2	3	4	5	6	7
Very Unfaithful	Unfaithful	Fairly Unfaithful	Neutral	Fairly Faithful	Faithful	Very Faithful

- How much confidence do you have in the quality of your final plan, with respect to the mission description?

1	2	3	4	5	6	7
Very Unconfident	Unconfident	Fairly unconfident	Neutral	Fairly Confident	Confident	Very Confident

- How much do you trust the HGA testbed?

(Trust: The attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability)

1	2	3	4	5	6	7
Very Untrustworthy	Untrustworthy	Fairly untrustworthy	Neutral	Fairly Trustworthy	Trustworthy	Very Trustworthy

The user's understanding of the algorithm and strategy was evaluated with the following questions:

- Rate how the HGA testbed helped support your understanding of the algorithm's logic (understanding of how algorithm works, what it does, why it does it, etc.):

1	2	3	4	5	6	7
Very Unhelpful	Unhelpful	Fairly Unhelpful	Neutral	Fairly Helpful	Helpful	Very Helpful

- How would you describe your strategy for finding the best plan?

The utility of the HGA testbed was evaluated with the following questions:

- Rate how the HGA testbed helped support your ability to guide the algorithm to produce the plans you wanted:

1	2	3	4	5	6	7
Very Unhelpful	Unhelpful	Fairly Unhelpful	Neutral	Fairly Helpful	Helpful	Very Helpful

- How would you rate the difficulty in operating the HGA testbed?

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

The Sliding Bars, Pie Chart Control, and Pie Chart Control with Sensitivity Analysis Display were evaluated with the following questions:

- Rate how difficult it was to use the TBD control:

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

- Explain how you used the TBD control:
- Did the TBD control have an effect on your strategy for finding the best plan?
- Did the TBD tool support/detract from your understanding of the algorithm's logic? How?
- Rate how useful the TBD tool was in supporting your decision making:

1	2	3	4	5	6	7
Very Useless	Useless	Fairly Useless	Neutral	Fairly Useful	Useful	Very Useful

The rationale window, plan cycle option, metric history, and weight history were evaluated with the following questions:

- Did you use the TBD display during this scenario?
- Rate how difficult it was to use the TBD display:

1	2	3	4	5	6	7
Very Easy	Easy	Fairly Easy	Neutral	Fairly Difficult	Difficult	Very Difficult

- Explain how you used the TBD display:

- Did the TBD display affect your strategy?

- Did the TBD display improve/support/detract from your understanding of the algorithm's logic? How?

- Rate how useful the TBD display was in supporting your decision making:

1	2	3	4	5	6	7
Very Useless	Useless	Fairly Useless	Neutral	Fairly Useful	Useful	Very Useful

The post experiment questionnaire probed issues of trust in the HGA's individual components, utility of the TBD tools, user desire for trust information, and user's overall preference of the HGA testbed or global search algorithm:

- Were there system components (controls, windows, metrics, algorithm, etc.) that you trusted more than others? Which ones? Why? Was there anything that caused confusion, lack of trust, etc?

  Specific: System Components:

Level Off:	Sliding Bars
Level 1:	Pie Chart
	Rationale Window
	Plan Cycle Option
Level 2:	Pie Chart w/ Sensitivity Analysis
	Weight History
	Metric History

- Would the Pie Chart be better if you could adjust the pie slices with the mouse? Why?

- If you were told that there was a high probability that the Expected Value calculation was wrong, would that change how you selected the final plan? How? Why?

- Did knowing that there was some variability in the target locations change your strategy of selecting a good plan? How?

- If you knew there were other inaccuracies or variability in the data (target locations, # of UAVs available, threats, time windows, etc) would you change how you selected the final plan? How?

- Would knowing the magnitude of the data inaccuracy or variability affect your decision making? How?

- If you had one more scenario to complete, but could only implement the HMCDM search or the computer search, which one would you pick? Why?

- If you had to do an HMCDM search, which tools would you want? Please check the boxes of the various components:

  - ☐ Sliding Bars or ☐ Pie Chart
  - ☐ Rationale Window
  - ☐ Plan Cycle Option
  - ☐ Sensitivity Analysis
  - ☐ Weight History
  - ☐ Metric History

Appendix B

DSV Plots

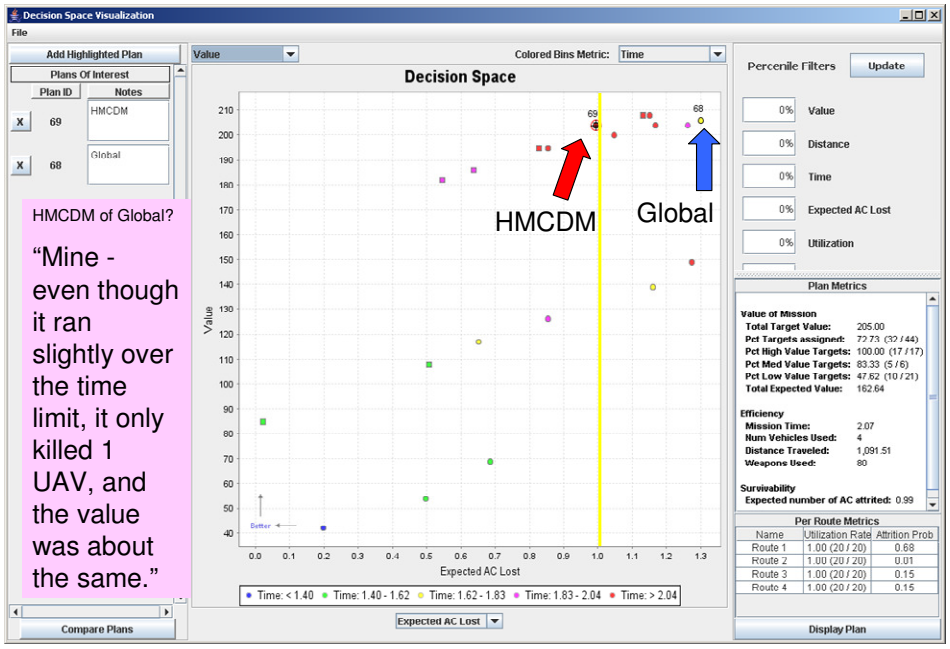
The vertical axis of the plot is either value or expected value, depending on what was inferred from the questionnaire as the user's preference in comparing plans. The horizontal axis is expected aircraft attrition. Colored markers are used to plot the plans. The color of the markers is dependent on the plan's time to complete the mission. A color scale is located beneath the chart. These metrics were selected because they were emphasized in the scenarios and the users commented that they used these metrics to evaluate plan quality. The default shape of the markers is a circle. The markers are square if they are on the local expected attrition/expected value or value Pareto frontier. A vertical yellow band marks the point on the graph where plans to the right of the line violate the constraint of losing more than one UAV. Plans with a number above them have comments in the left panel of the interface. An orange circle surrounds plans that *might* have been better than the user selected HMCDM plan. The comment in the bottom left panel explains the participant's rationale for selecting the number one ranked plan.

For each set of scenario 4 rankings, plans labeled with an 'H' are HMCDM plans and those with a 'G' are global plans. The number following the 'G' indicates the number of times the participant used the global search. For example, 'G1' indicates that the plan is the first time the participant ever used the global search. 'G3' indicates that the plan is the third and final time the participant used the global search. The purpose of the number is to determine if experience with the global search yielded improved global plans. The number above the plans reveals the ranking of that plan.

The plots are labeled P*X* Level *Y,* meaning participant *X* level *Y*. For example, P7 Level 2 is participant 7 Level 2 of TBD.

P1: Level 0

Value

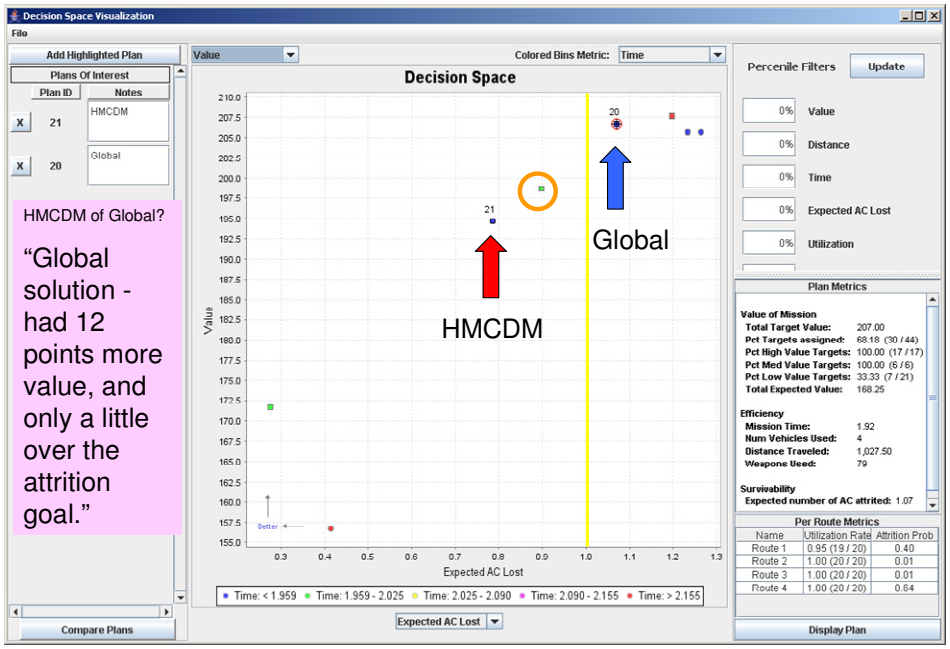


HMCDM of Global?

"Mine - even though it ran slightly over the time limit, it only killed 1 UAV, and the value was about the same."

P1: Level 1

Value

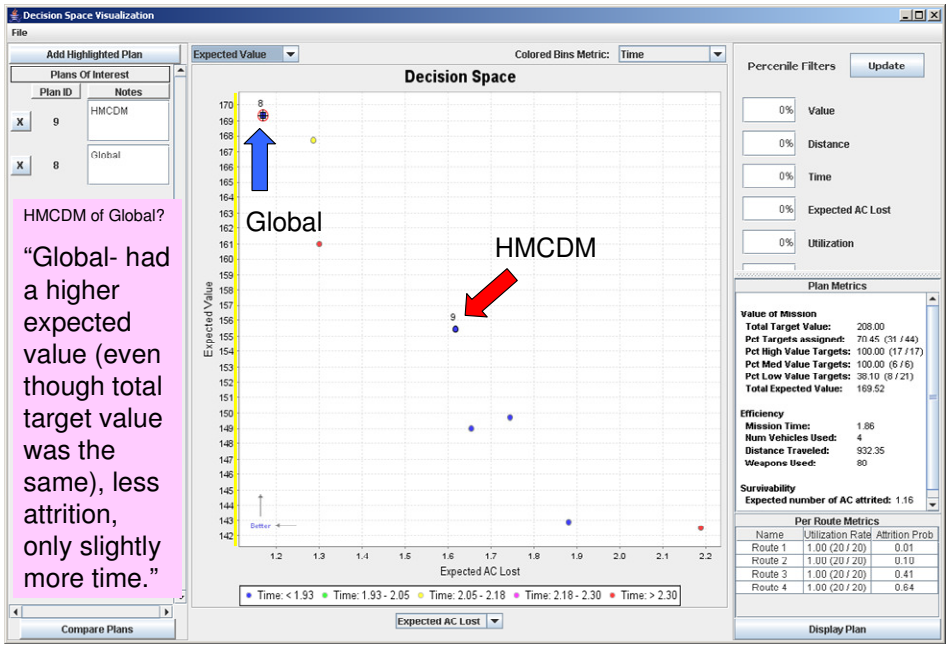


HMCDM of Global?

"Global solution - had 12 points more value, and only a little over the attrition goal."

P1: Level 2

Expected Value

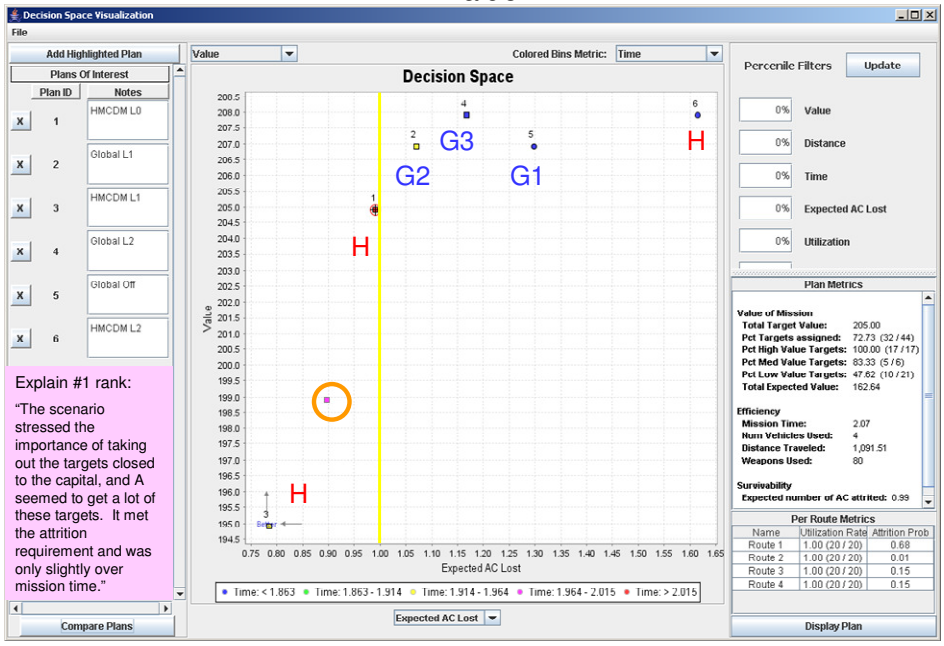


HMCDM of Global?

"Global- had a higher expected value (even though total target value was the same), less attrition, only slightly more time."

P1: Rankings

Value

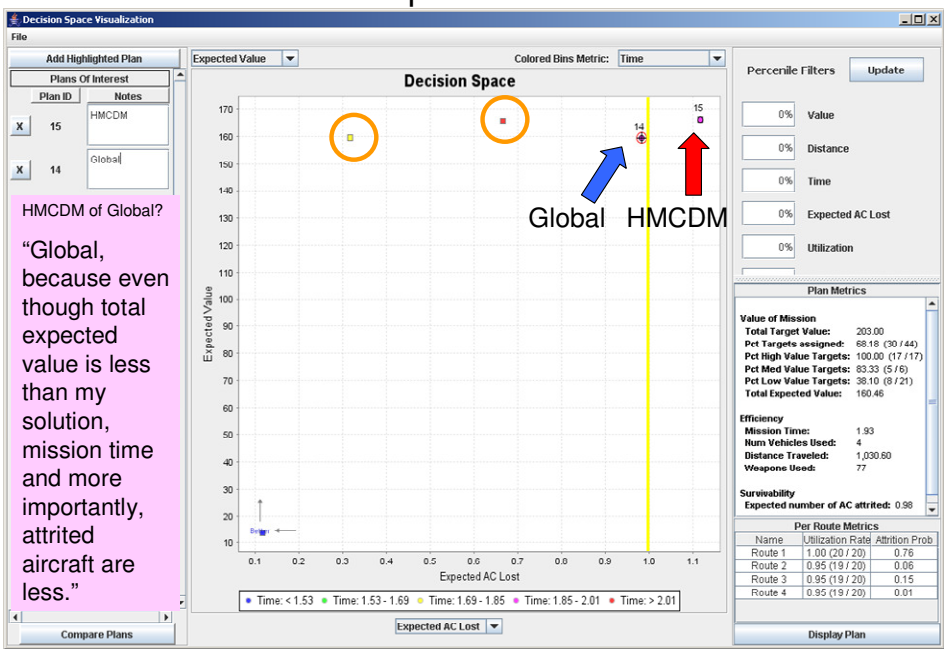


Explain #1 rank:

"The scenario stressed the importance of taking out the targets closed to the capital, and A seemed to get a lot of these targets. It met the attrition requirement and was only slightly over mission time."

P2: Level 0

Expected Value

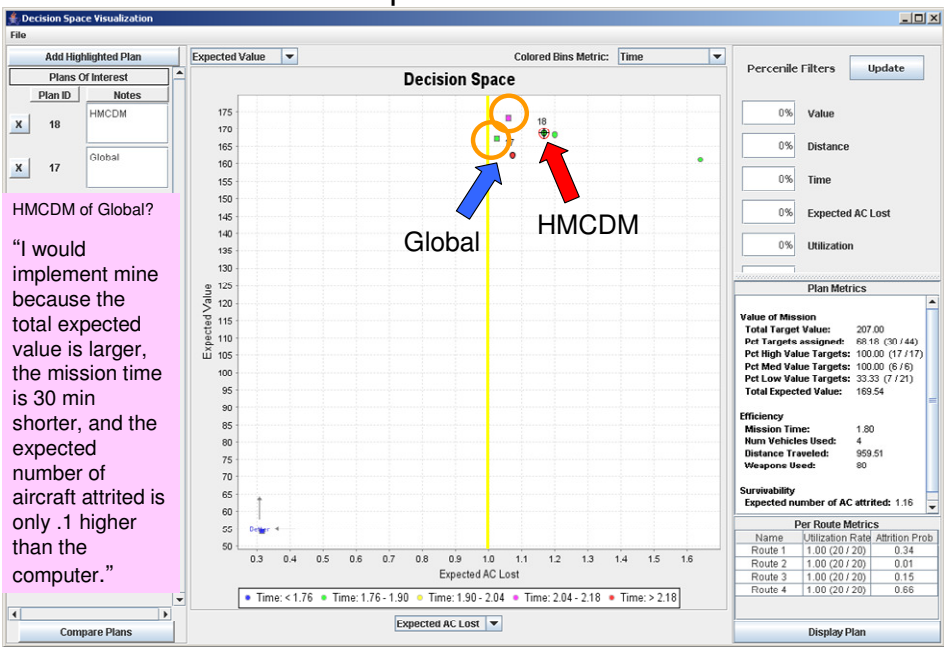


HMCDM of Global?

"Global, because even though total expected value is less than my solution, mission time and more importantly, attrited aircraft are less."

P2: Level 1

Expected Value

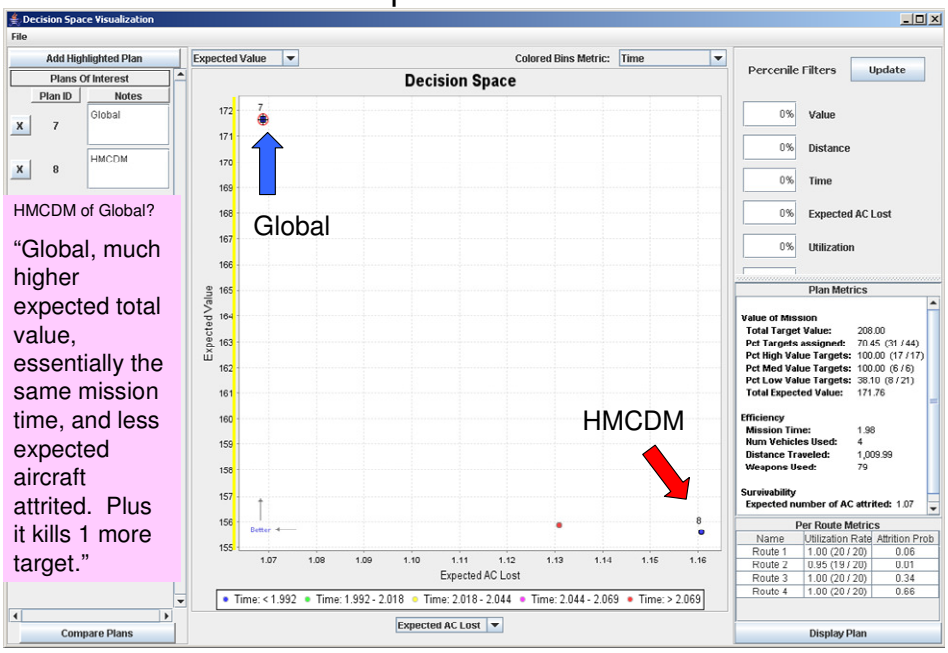


HMCDM of Global?

"I would implement mine because the total expected value is larger, the mission time is 30 min shorter, and the expected number of aircraft attrited is only .1 higher than the computer."

P2: Level 2

Expected Value

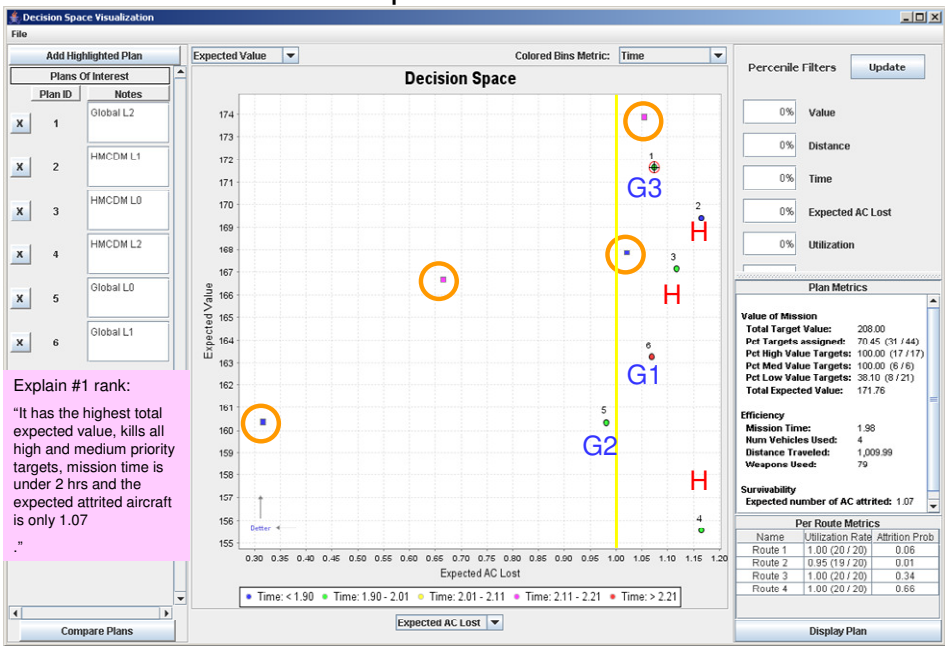


HMCDM of Global?

"Global, much higher expected total value, essentially the same mission time, and less expected aircraft attrited. Plus it kills 1 more target."

P2: Rankings

Expected Value



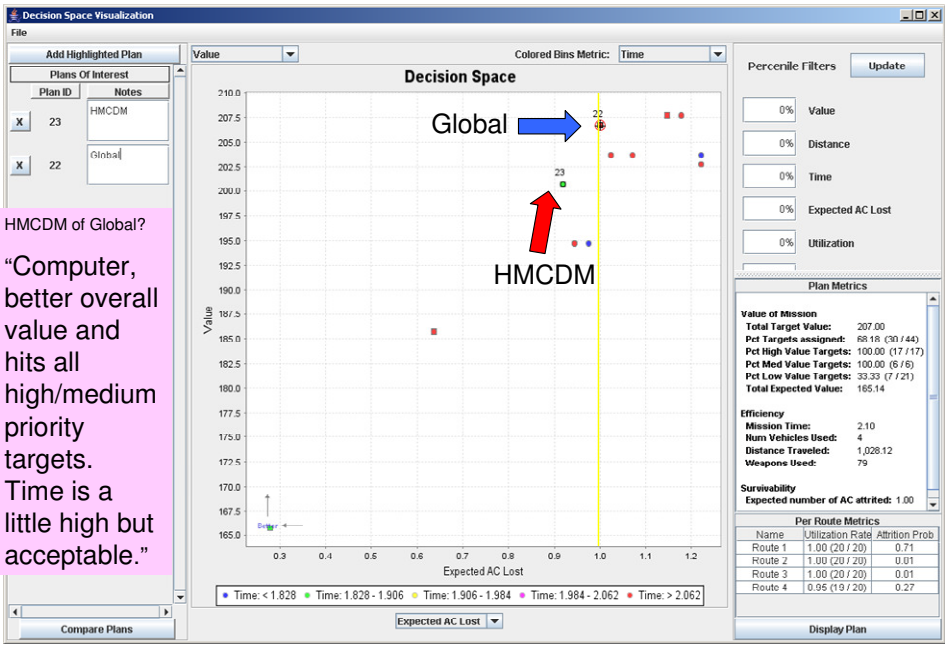
Explain #1 rank:

"It has the highest total expected value, kills all high and medium priority targets, mission time is under 2 hrs and the expected attrited aircraft is only 1.07

."

P3: Level 0

Value

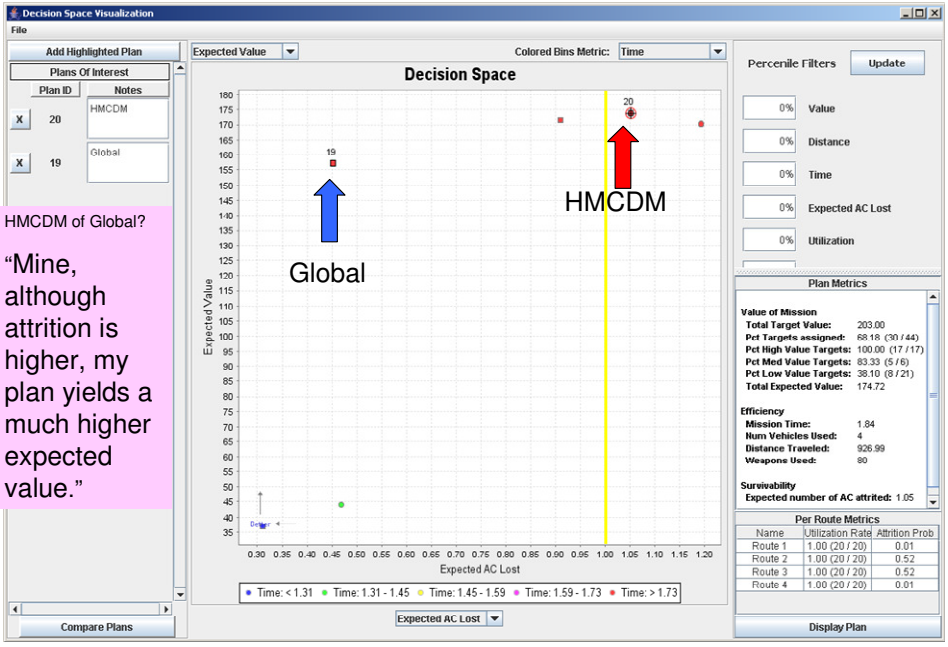


HMCDM of Global?

"Computer, better overall value and hits all high/medium priority targets. Time is a little high but acceptable."

P3: Level 1

Expected Value

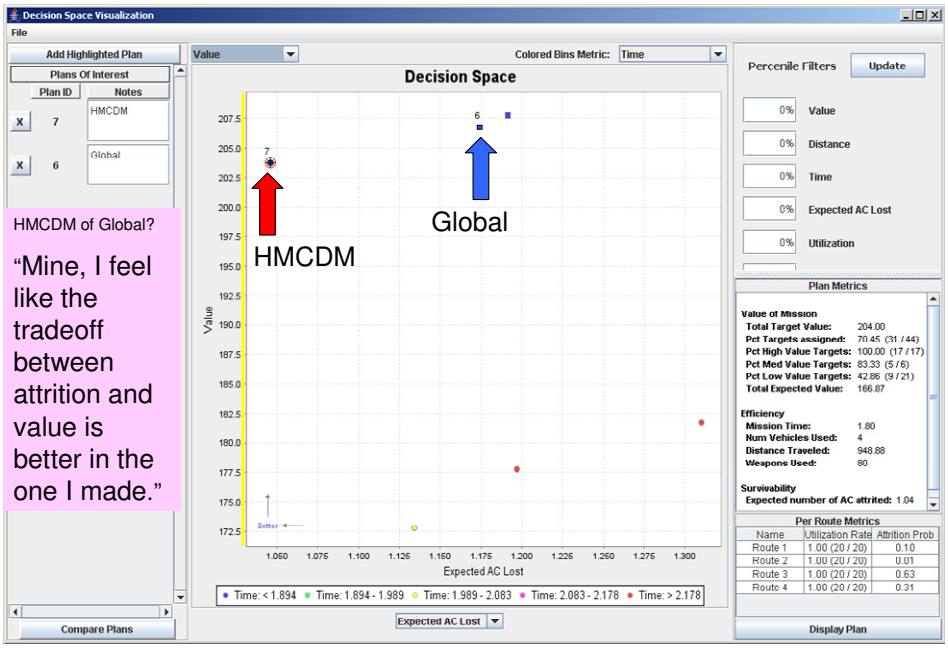


HMCDM of Global?

"Mine, although attrition is higher, my plan yields a much higher expected value."

P3: Level 2

Value

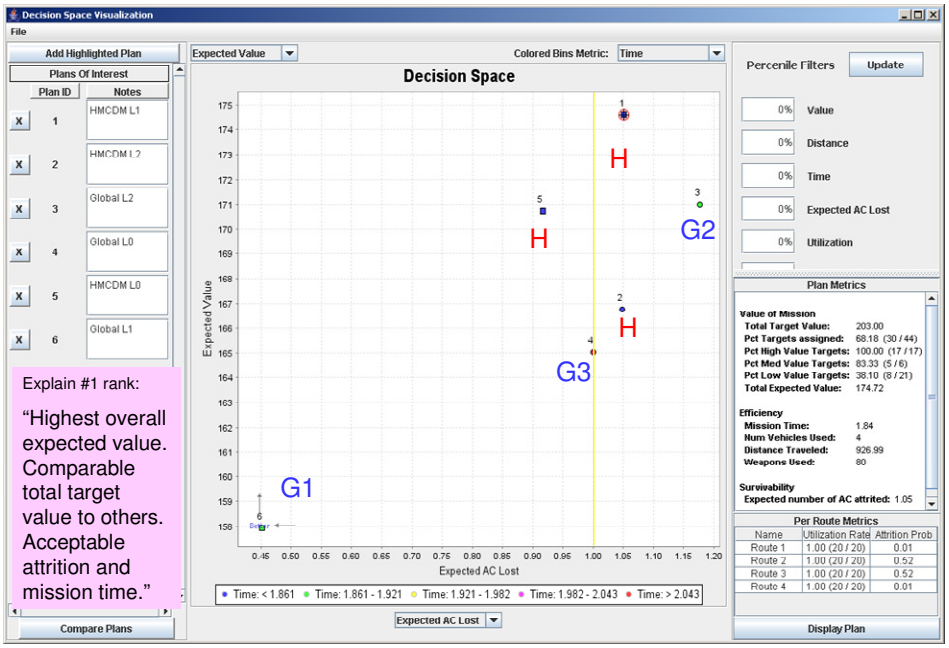


HMCDM of Global?

"Mine, I feel like the tradeoff between attrition and value is better in the one I made."

P3: Rankings

Expected Value

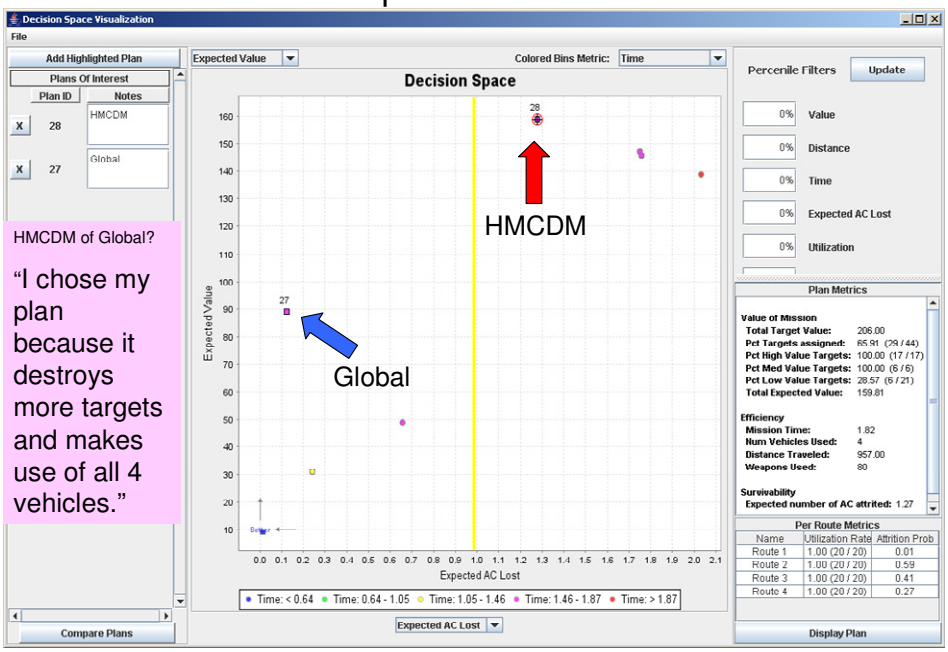


Explain #1 rank:

"Highest overall expected value. Comparable total target value to others. Acceptable attrition and mission time."

P4: Level 0

Expected Value

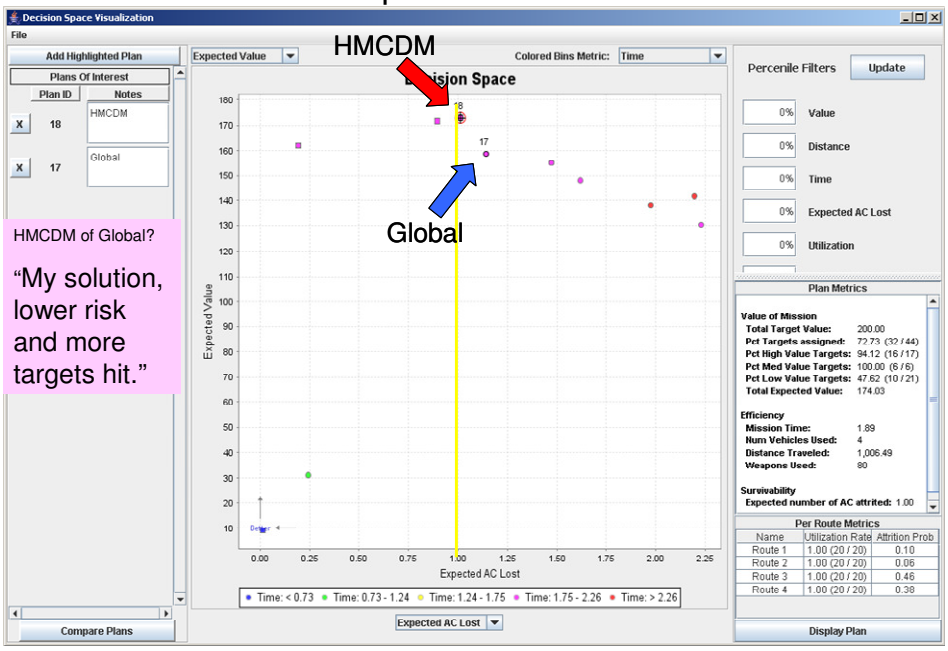


HMCDM of Global?

"I chose my plan because it destroys more targets and makes use of all 4 vehicles."

P4: Level 1

Expected Value

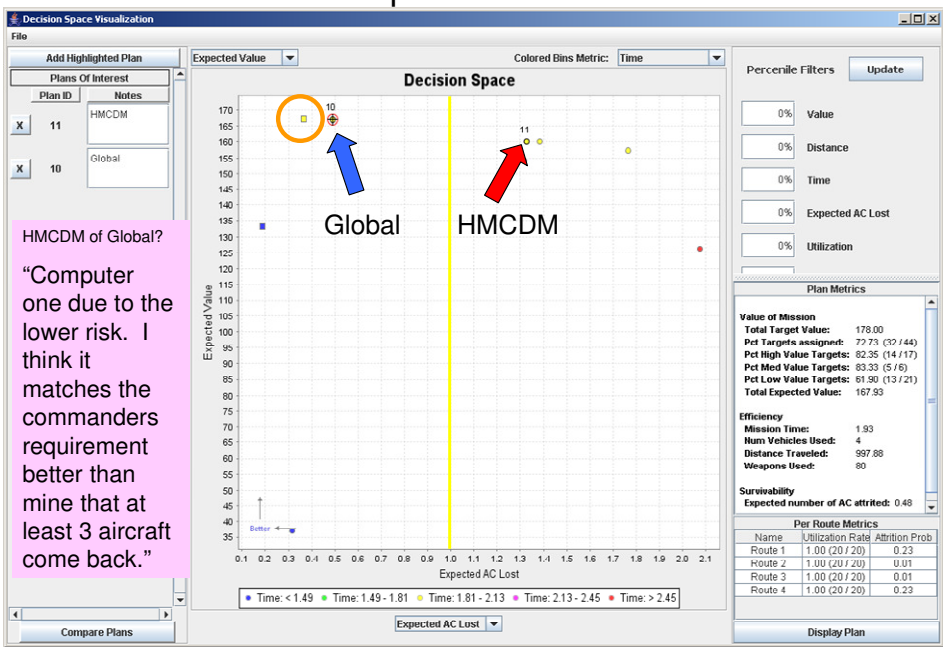


HMCDM of Global?

"My solution, lower risk and more targets hit."

P4: Level 2

Expected Value

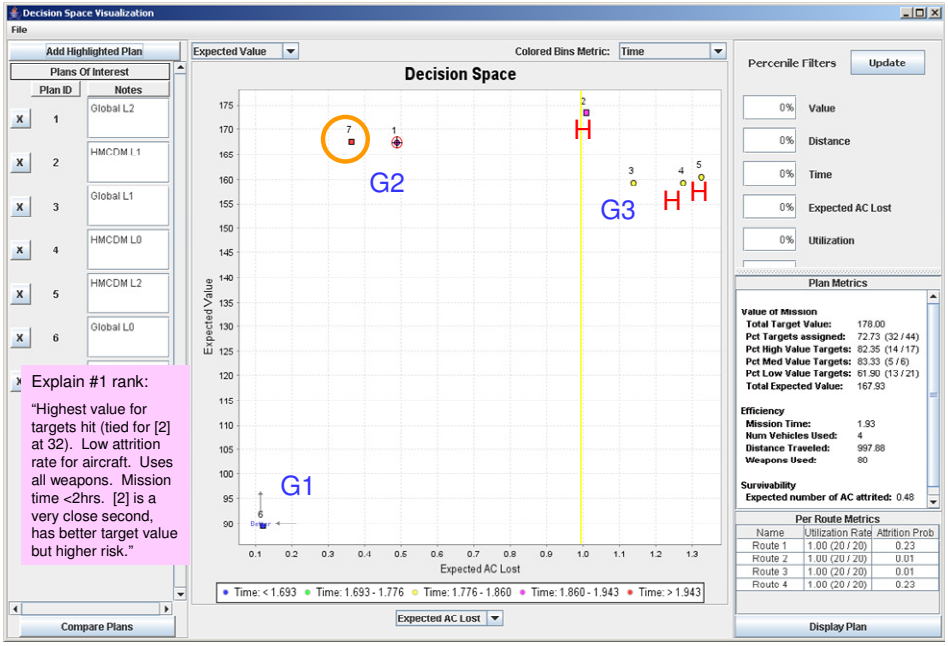


HMCDM of Global?

"Computer one due to the lower risk. I think it matches the commanders requirement better than mine that at least 3 aircraft come back."

P4: Rankings

Expected Value



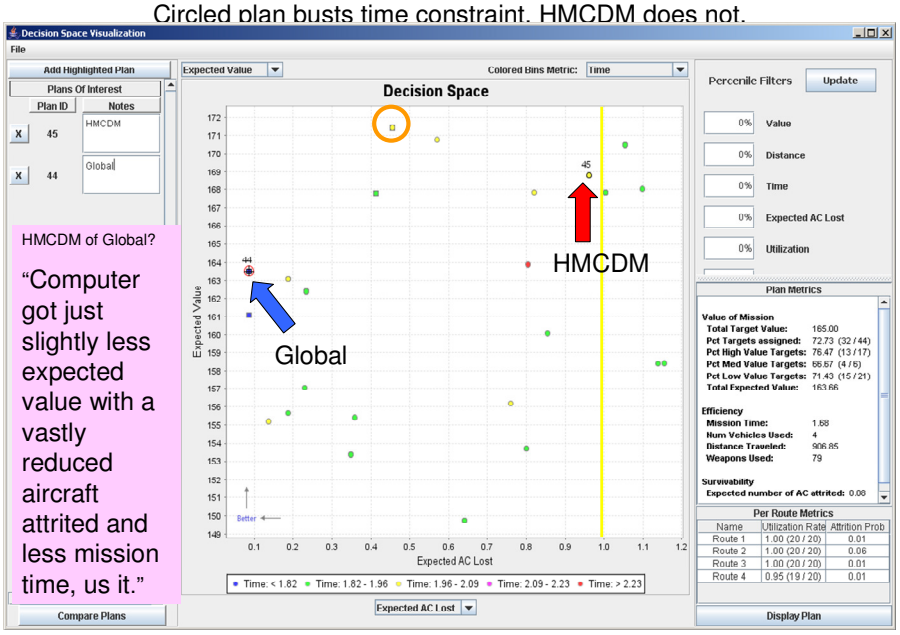
Explain #1 rank:

"Highest value for targets hit (tied for [2] at 32). Low attrition rate for aircraft. Uses all weapons. Mission time <2hrs. [2] is a very close second, has better target value but higher risk."

P5: Level 0

Expected Value

Circled plan busts time constraint. HMCDM does not.

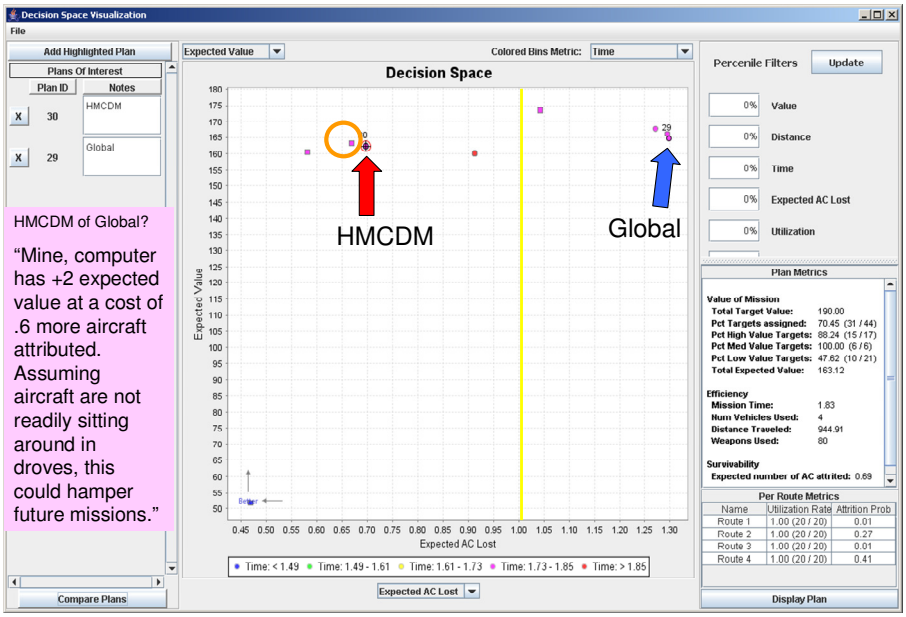


HMCDM of Global?

"Computer got just slightly less expected value with a vastly reduced aircraft attrited and less mission time, us it."

P5: Level 1

Expected Value

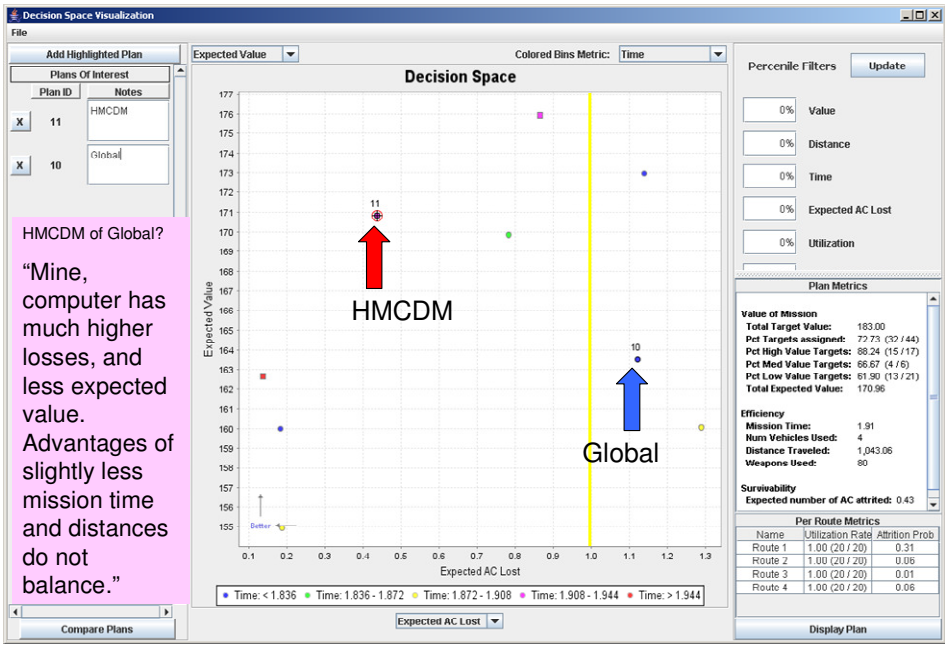


HMCDM of Global?

"Mine, computer has +2 expected value at a cost of .6 more aircraft attributed. Assuming aircraft are not readily sitting around in droves, this could hamper future missions."

P5: Level 2

Expected Value

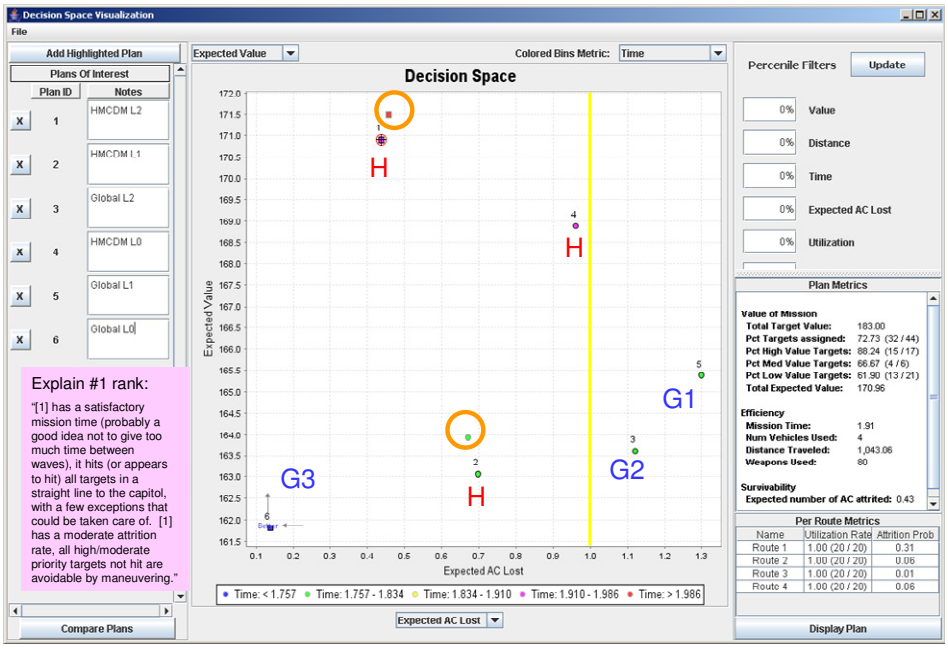


HMCDM of Global?

"Mine, computer has much higher losses, and less expected value. Advantages of slightly less mission time and distances do not balance."

P5: Ranking

Expected Value

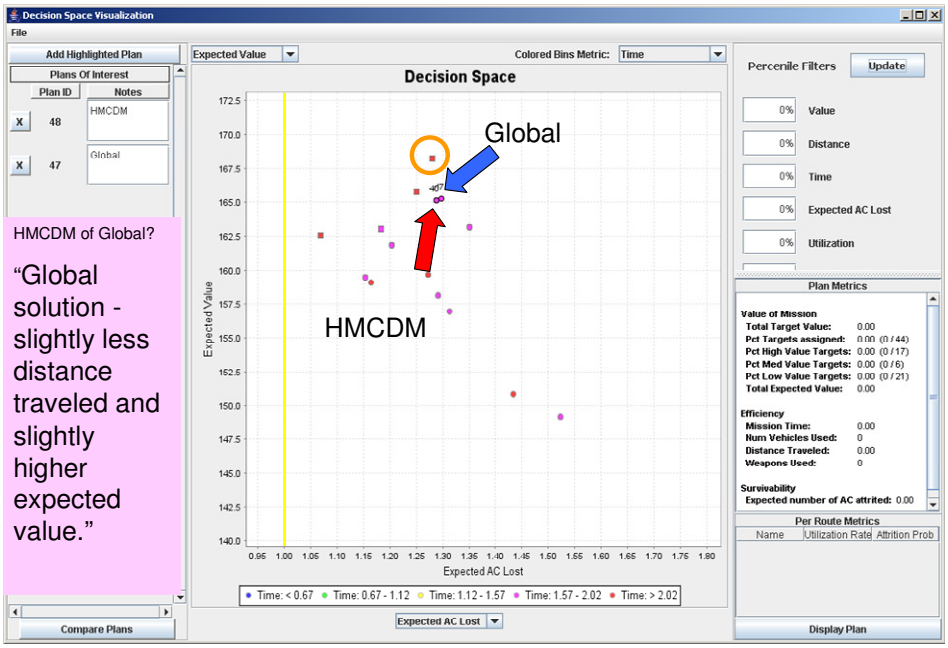


Explain #1 rank:

"[1] has a satisfactory mission time (probably a good idea not to give too much time between waves), it hits (or appears to hit) all targets in a straight line to the capitol, with a few exceptions that could be taken care of. [1] has a moderate attrition rate, all high/moderate priority targets not hit are avoidable by maneuvering."

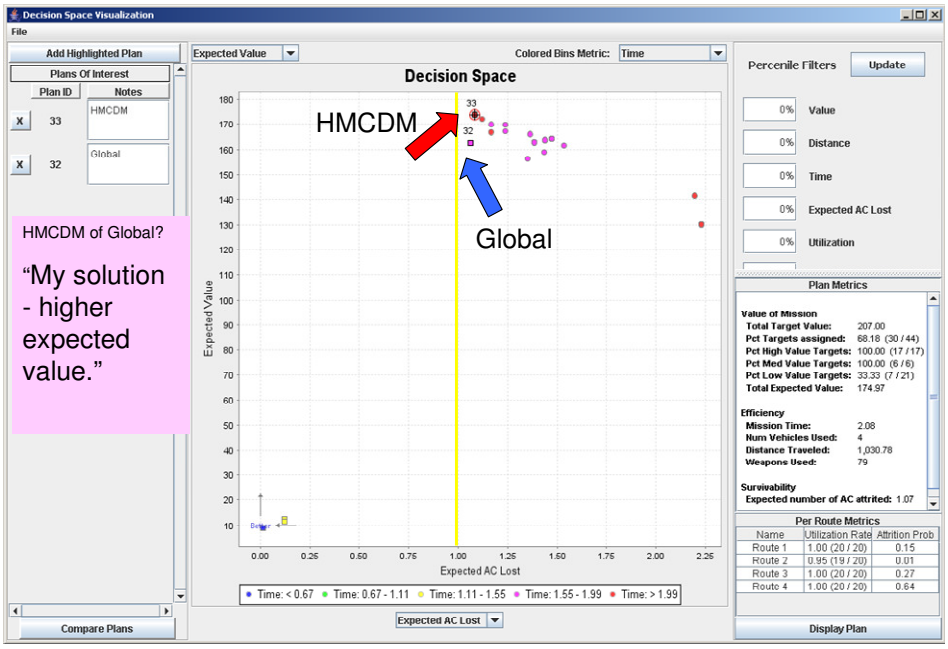
P6: Level 0

Expected Value: zoomed in



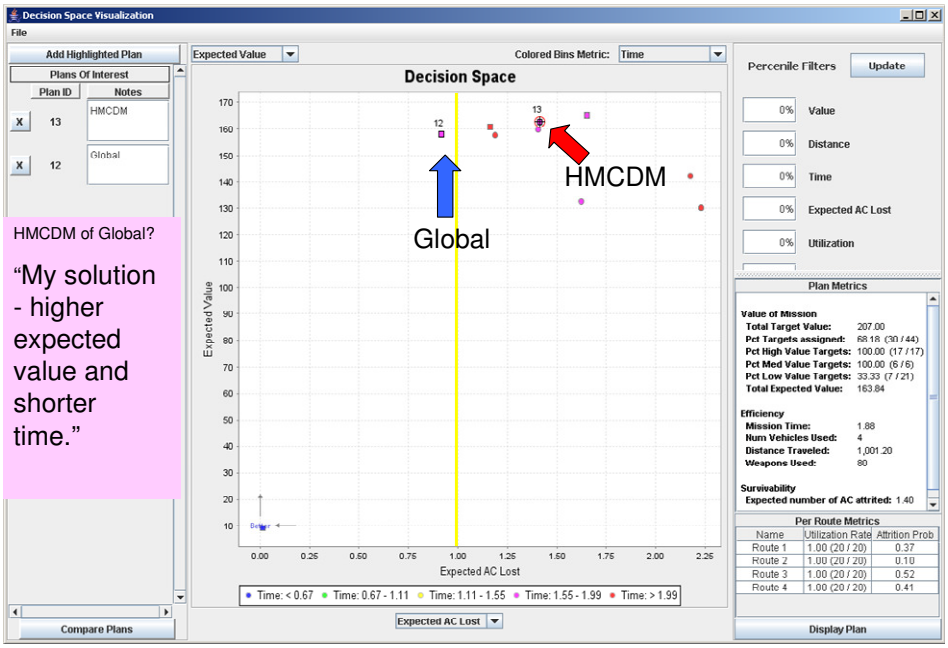
P6: Level 1

Expected Value



P6: Level 2

Expected Value

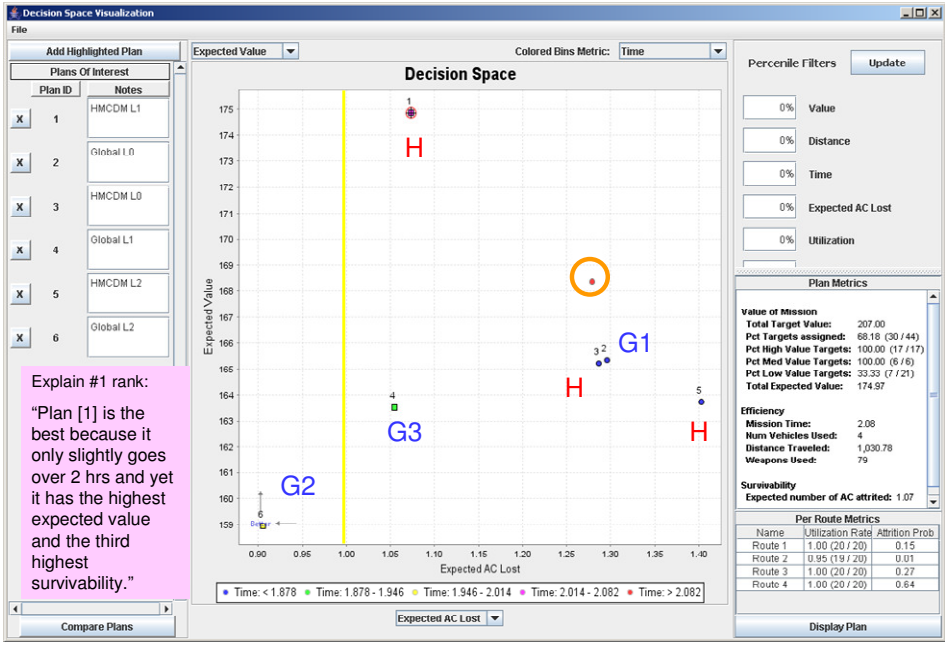


HMCDM of Global?

"My solution - higher expected value and shorter time."

P6: Rankings

Expected Value

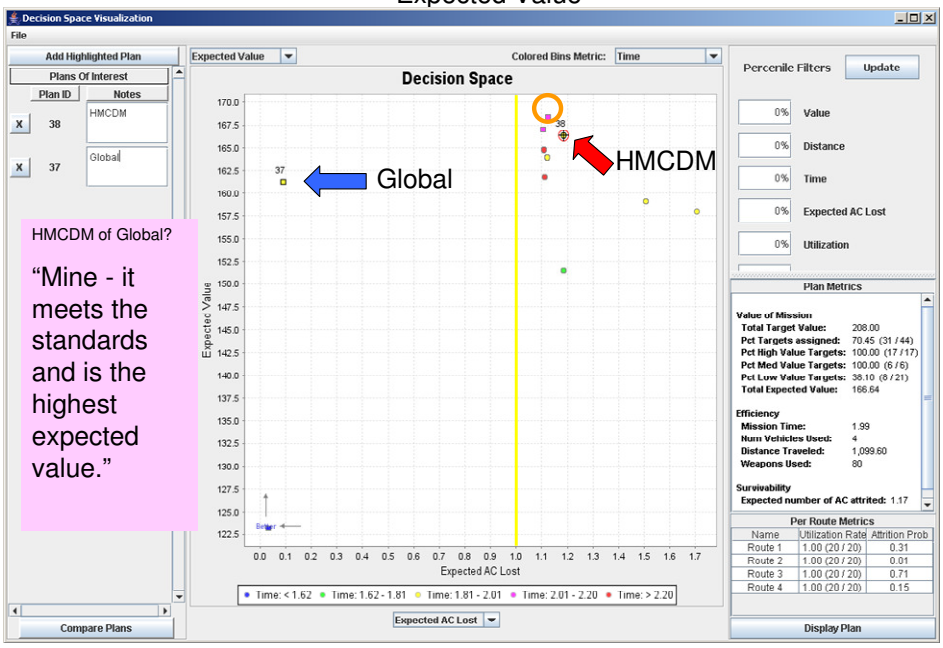


Explain #1 rank:

"Plan [1] is the best because it only slightly goes over 2 hrs and yet it has the highest expected value and the third highest survivability."

P7: Level 0

Expected Value

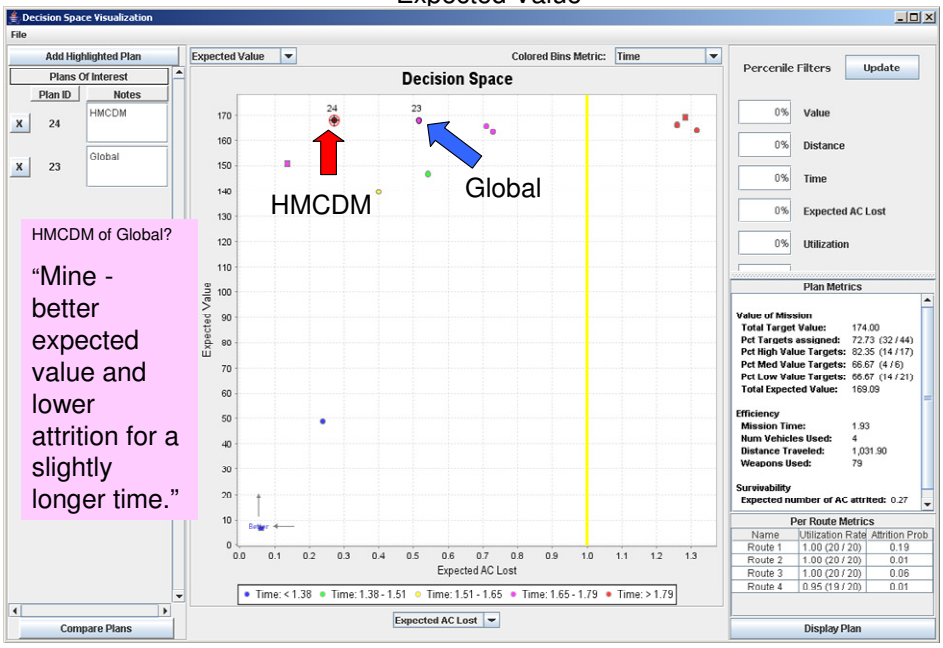


HMCDM of Global?

"Mine - it meets the standards and is the highest expected value."

P7: Level 1

Expected Value

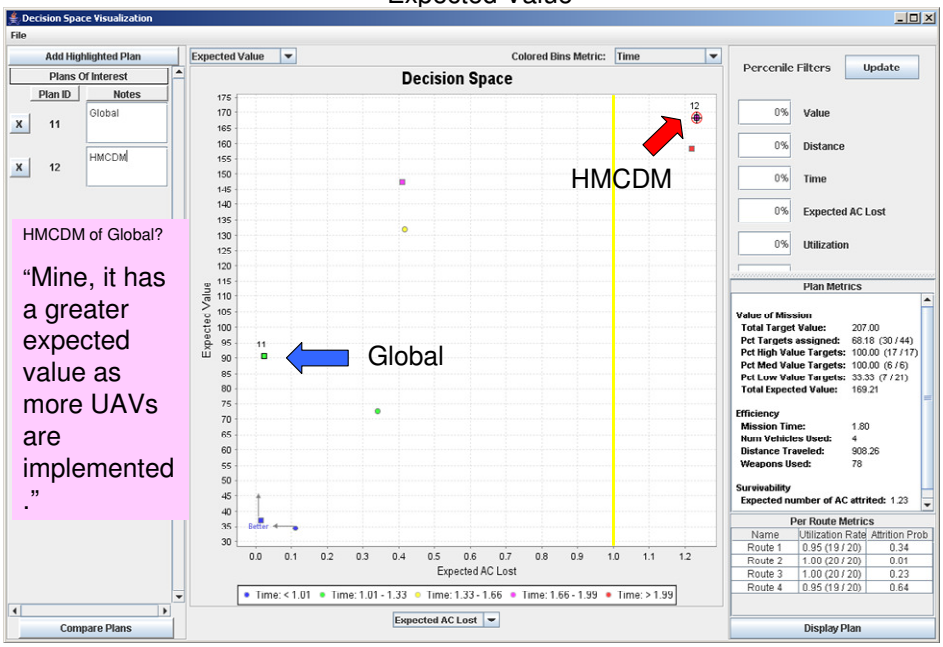


HMCDM of Global?

"Mine - better expected value and lower attrition for a slightly longer time."

P7: Level 2

Expected Value

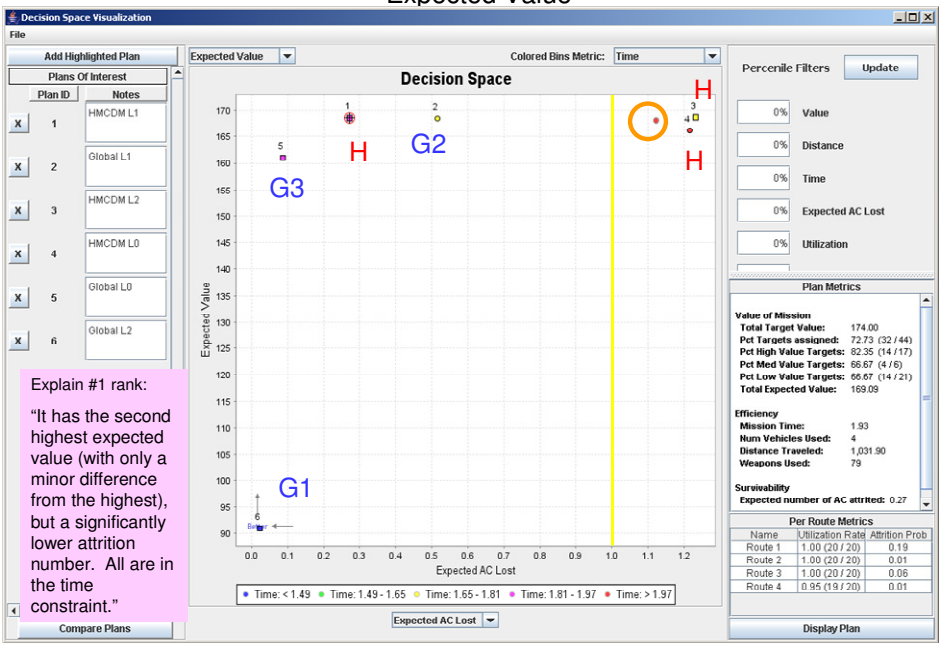


HMCDM of Global?

"Mine, it has a greater expected value as more UAVs are implemented ."

P7: Rankings

Expected Value

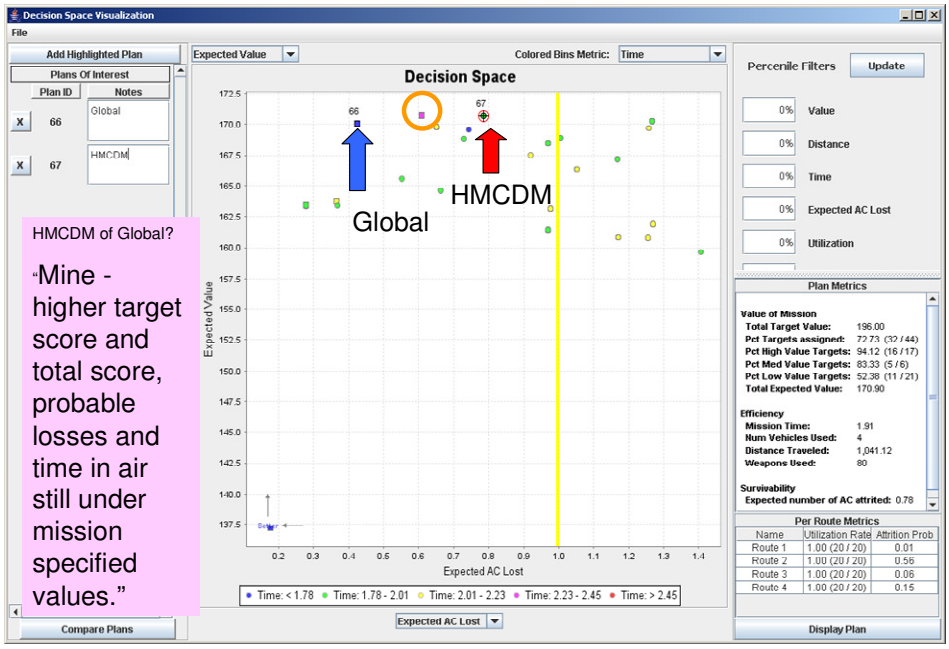


Explain #1 rank:

"It has the second highest expected value (with only a minor difference from the highest), but a significantly lower attrition number. All are in the time constraint."

P8: Level 0

Expected Value

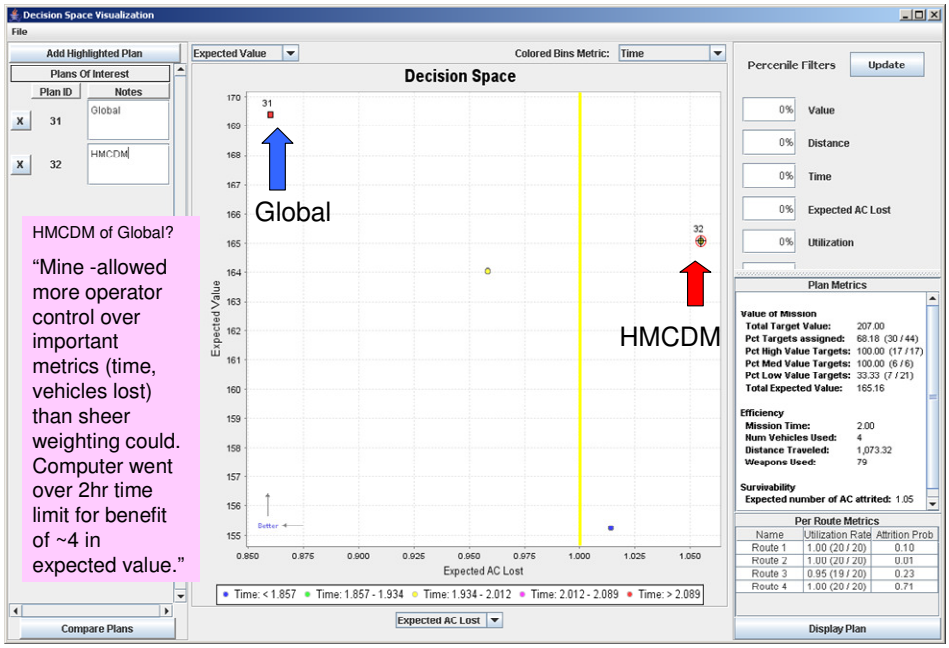


HMCDM of Global?

"Mine - higher target score and total score, probable losses and time in air still under mission specified values."

P8: Level 1

Expected Value

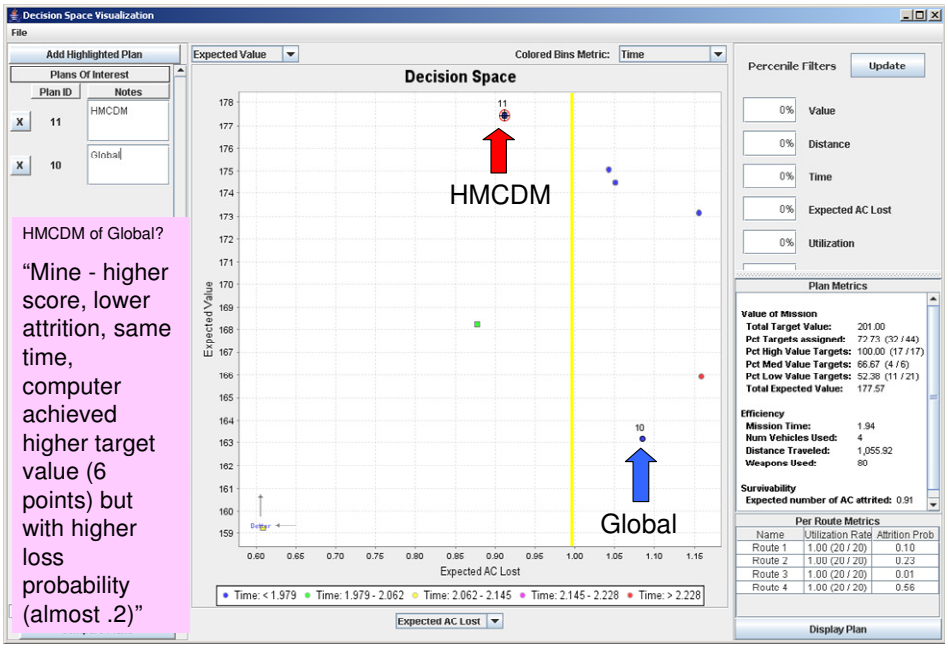


HMCDM of Global?

"Mine -allowed more operator control over important metrics (time, vehicles lost) than sheer weighting could. Computer went over 2hr time limit for benefit of ~4 in expected value."

P8: Level 2

Expected Value

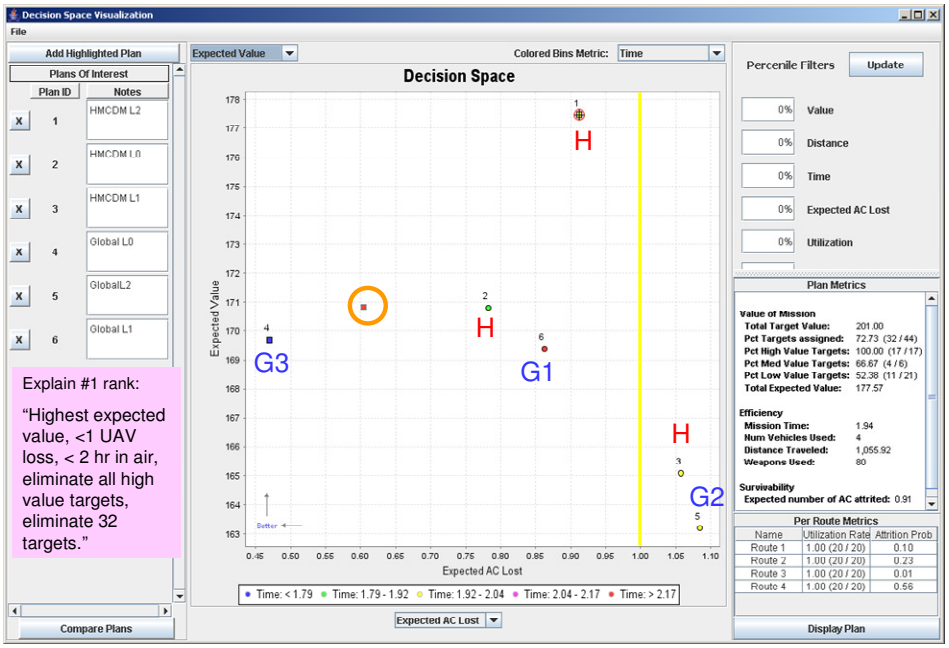


HMCDM of Global?

"Mine - higher score, lower attrition, same time, computer achieved higher target value (6 points) but with higher loss probability (almost .2)"

P8: Rankings

Expected Value



Explain #1 rank:

"Highest expected value, <1 UAV loss, < 2 hr in air, eliminate all high value targets, eliminate 32 targets."

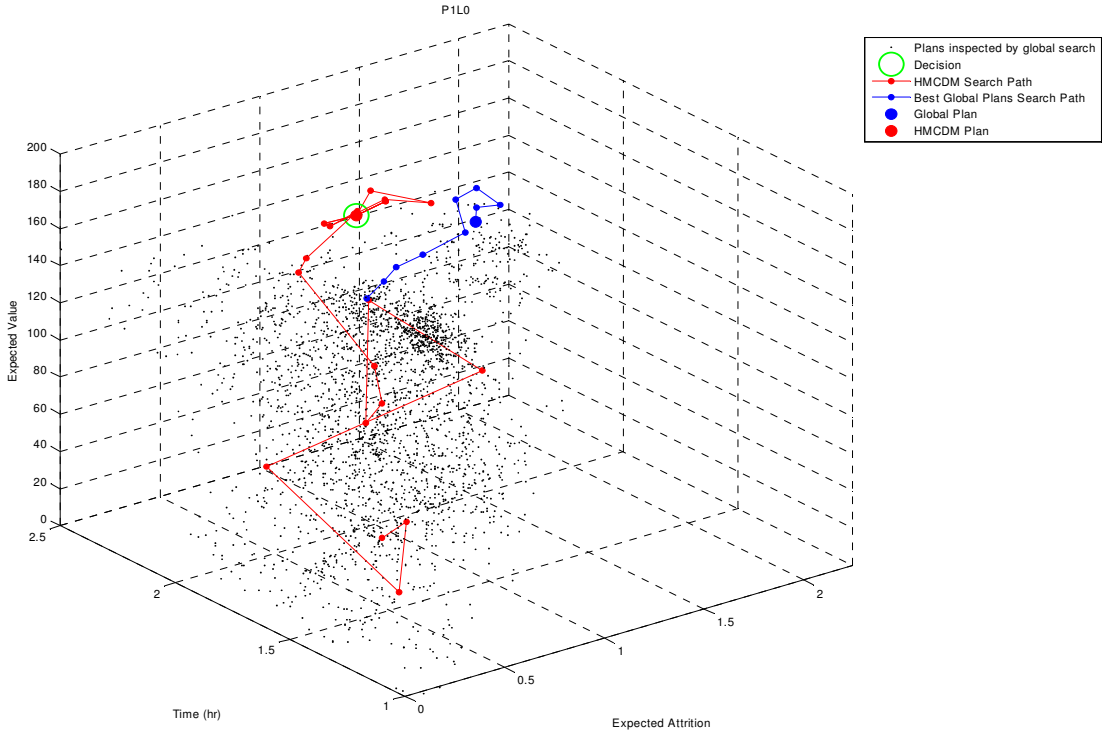
[This Page Intentionally Left Blank]

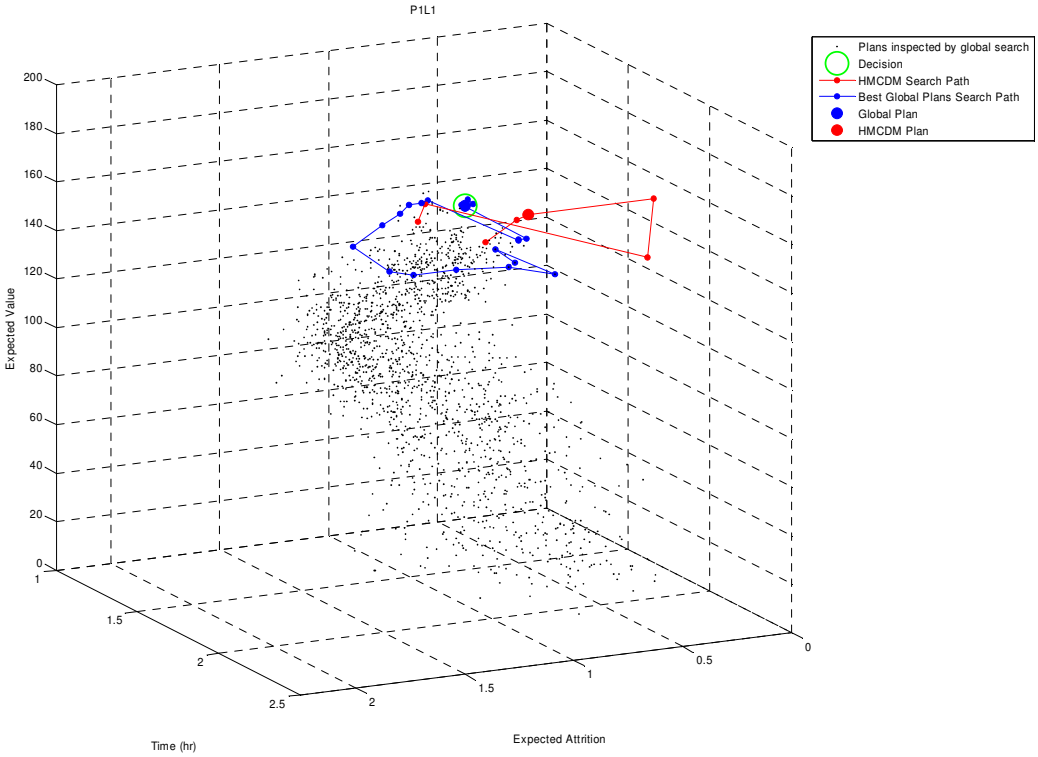
Appendix C

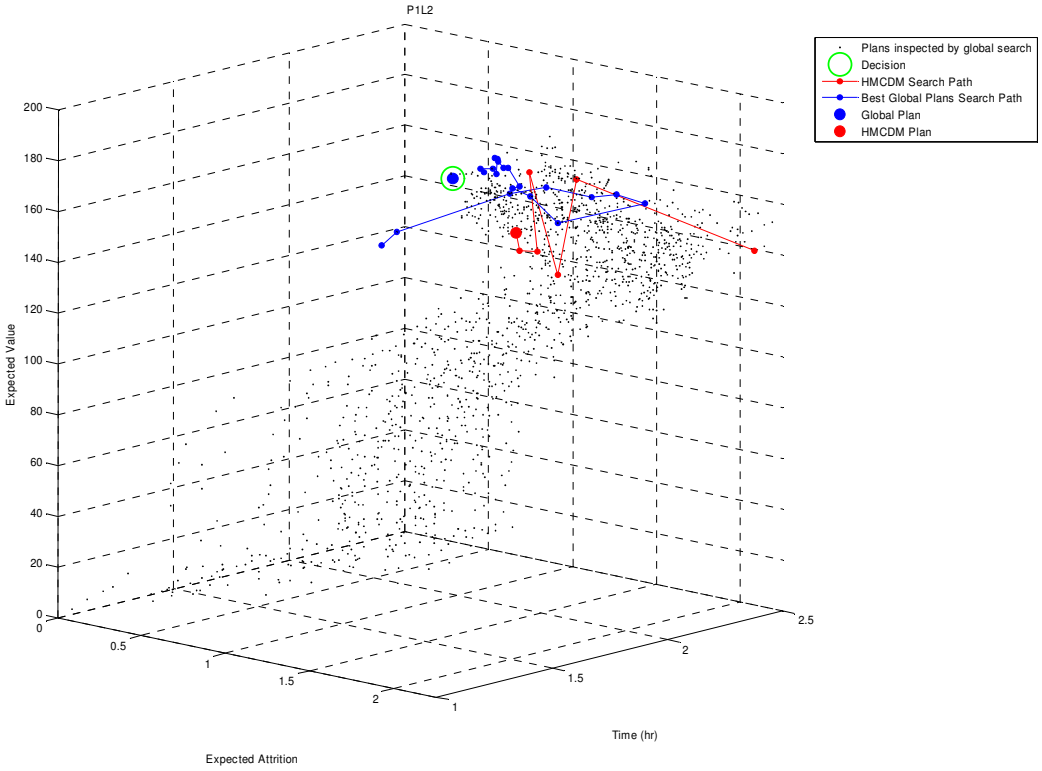
3D Plots

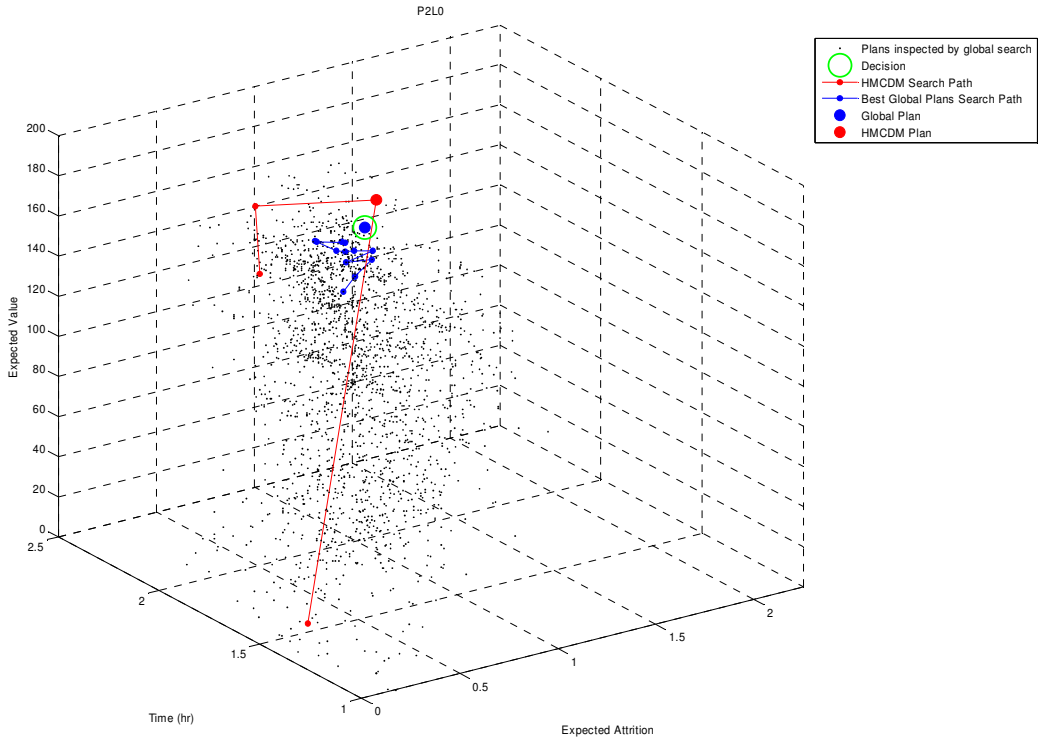
The axes were determined by the three metrics emphasized in each scenario: expected value, expected aircraft attrition, and time. The z-axis is expected value, and the x and y-axes are either time or expected attrition, depending on the graph's rotation. Each graph is rotated to a position that allows the best view of the plot. The solid red circles represent the plans generated by the HGA and they are connected with a red line to show the order in which they were generated. The large solid red circle is the HMCDM plan. During the experiment, only the search coefficients and the global plan were saved. Using these coefficients, the global search was run again after the experiment for 15 minutes and all of the plans were saved. The plans considered by the global algorithm in the post-experiment search are represented by black dots; these are the plans visited by the global search and do not include the neighborhoods of these plans that were also considered. The solid blue circles represent the best global plans as the algorithm searched the solution space during the post-experiment search. The blue line is the global plan search path, which shows the order that the best global plans were generated. The large solid blue circle represents the global plan. Since the global plan is the best global plan found during the experiment and the global plans on the plot were found after the experiment, the global plan might not be on the blue line. Finally, the participant-determined best plan for that scenario, either the global or HMCDM plan, is circled in green.

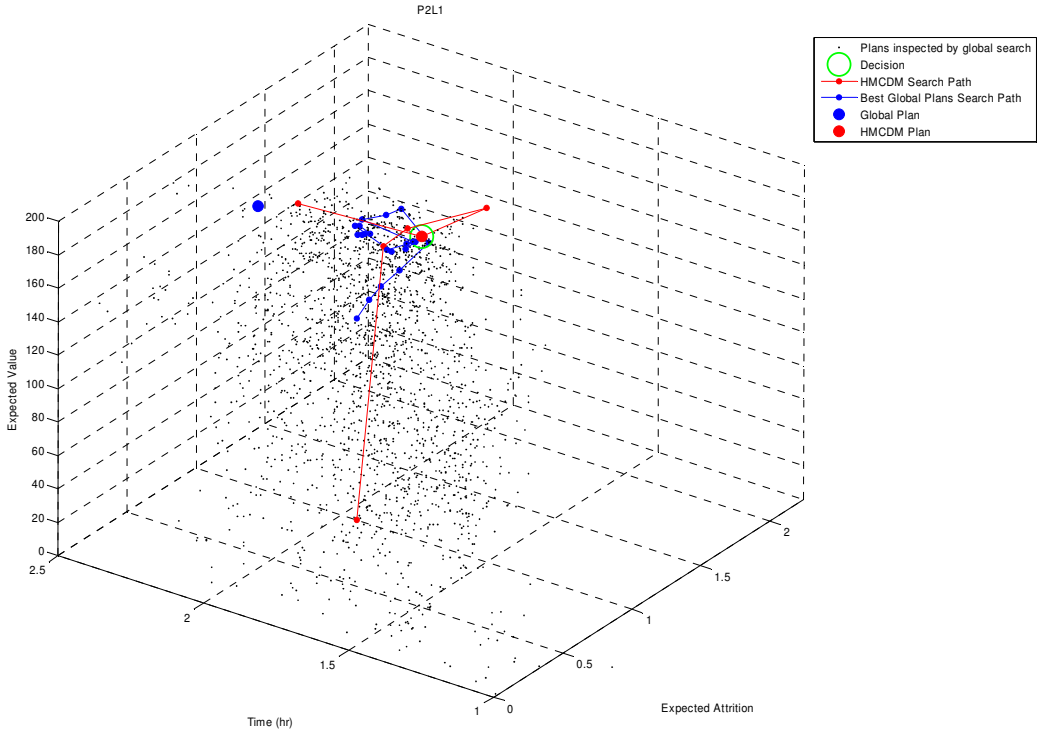
The plots are labeled P*X*L*Y,* meaning participant *X* level *Y*. For example, P7L2 is participant 7 Level 2 of TBD.

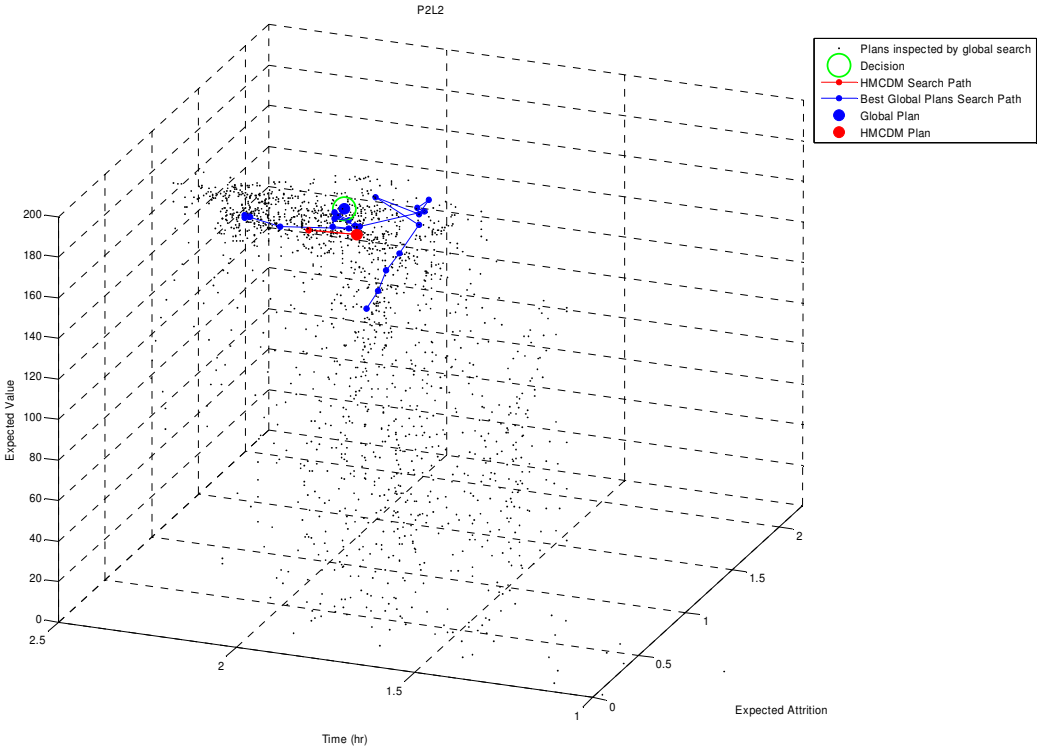

P1L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

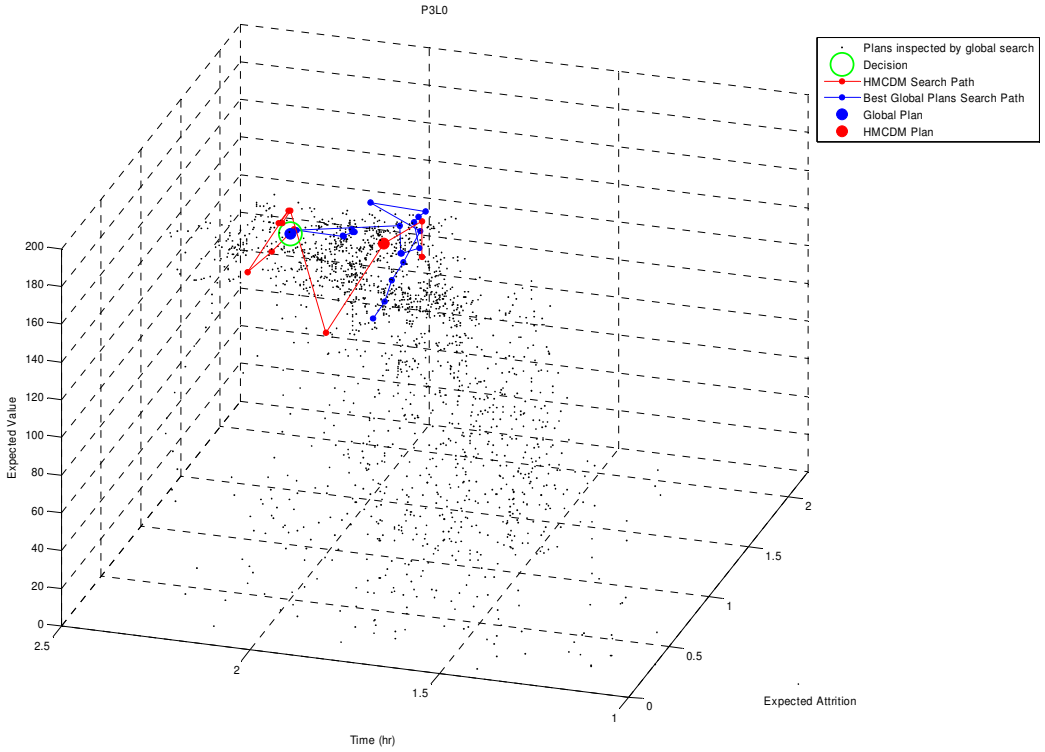

P1L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

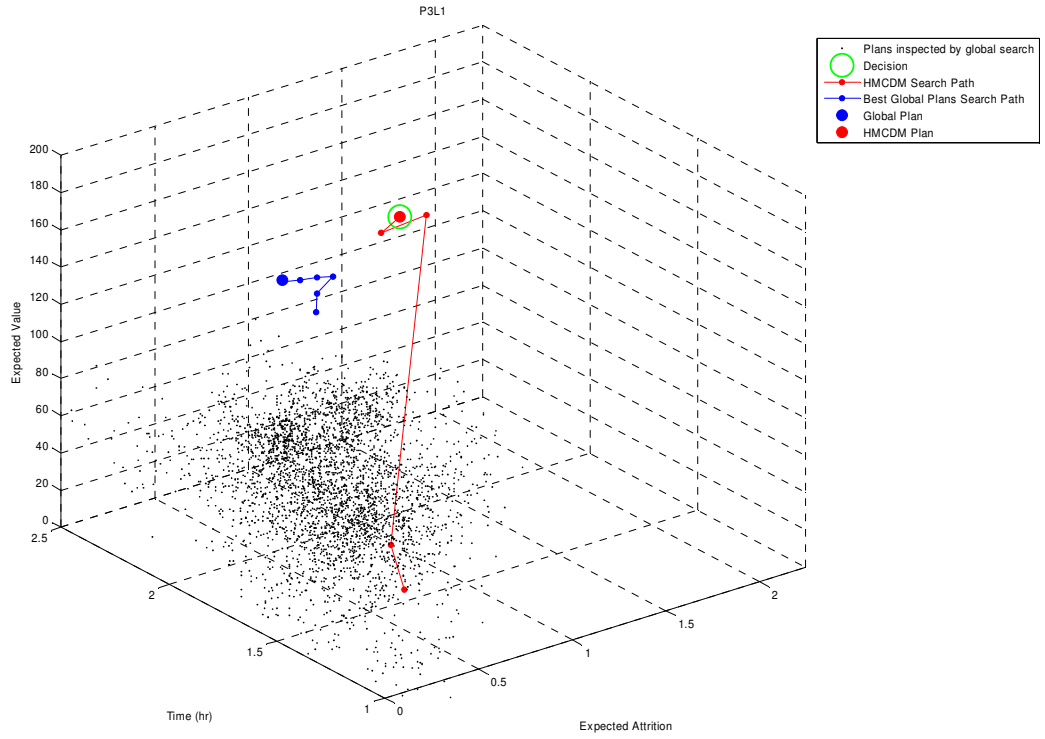

P1L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Expected Attrition
Time (hr)

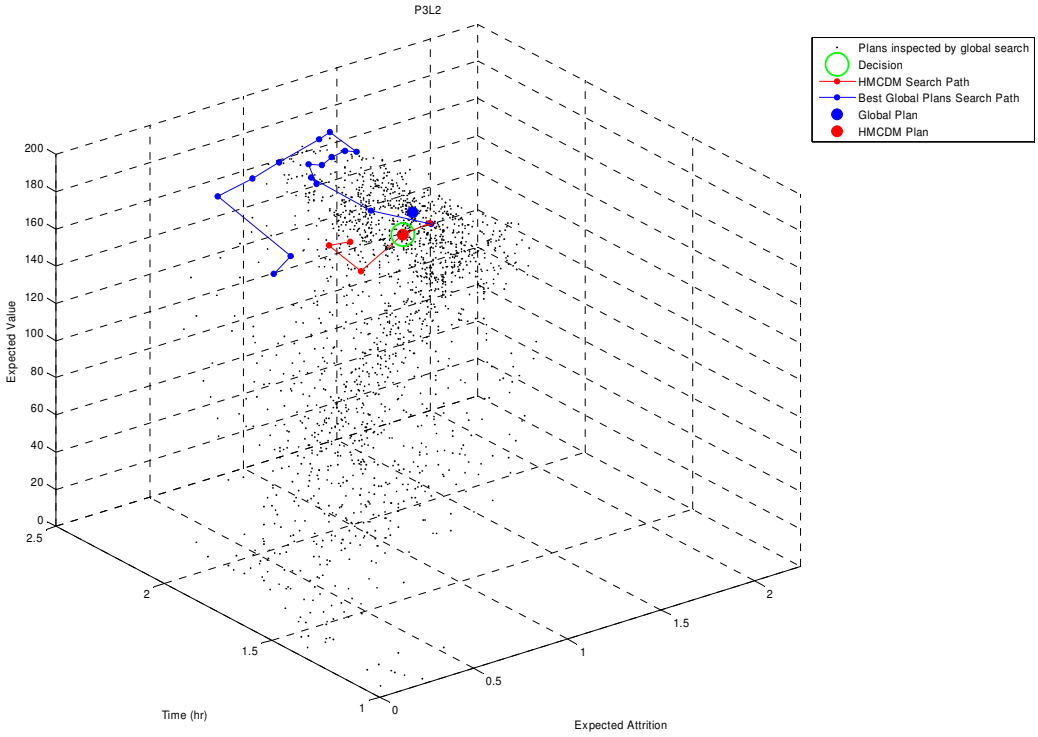

P2L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

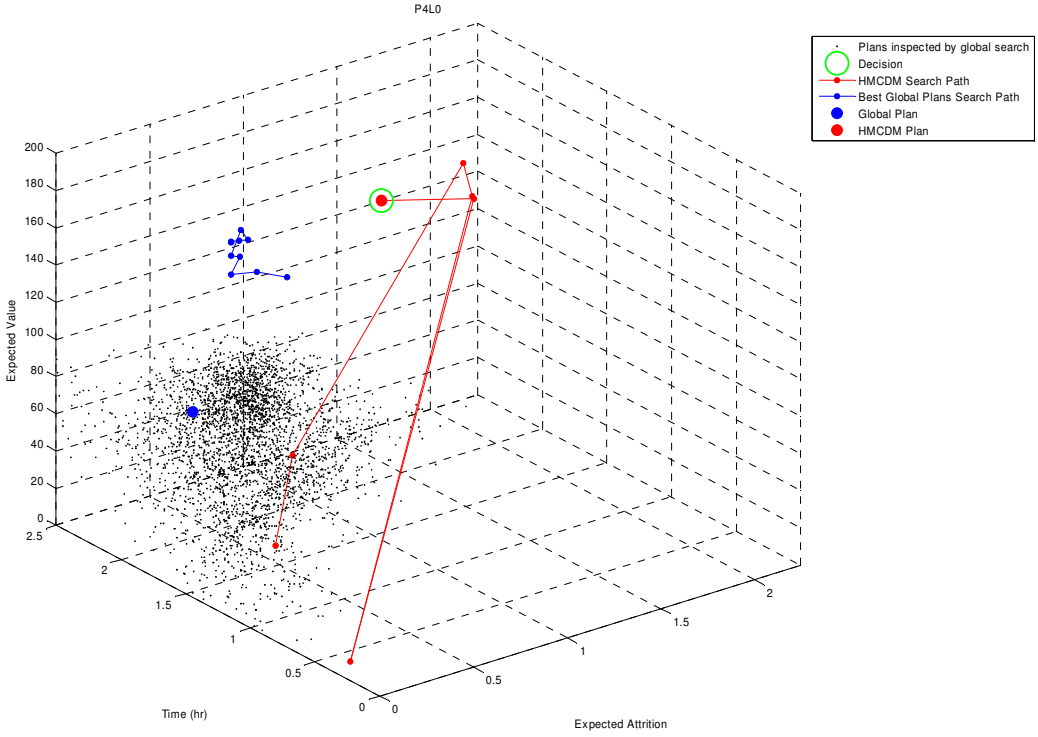

P2L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

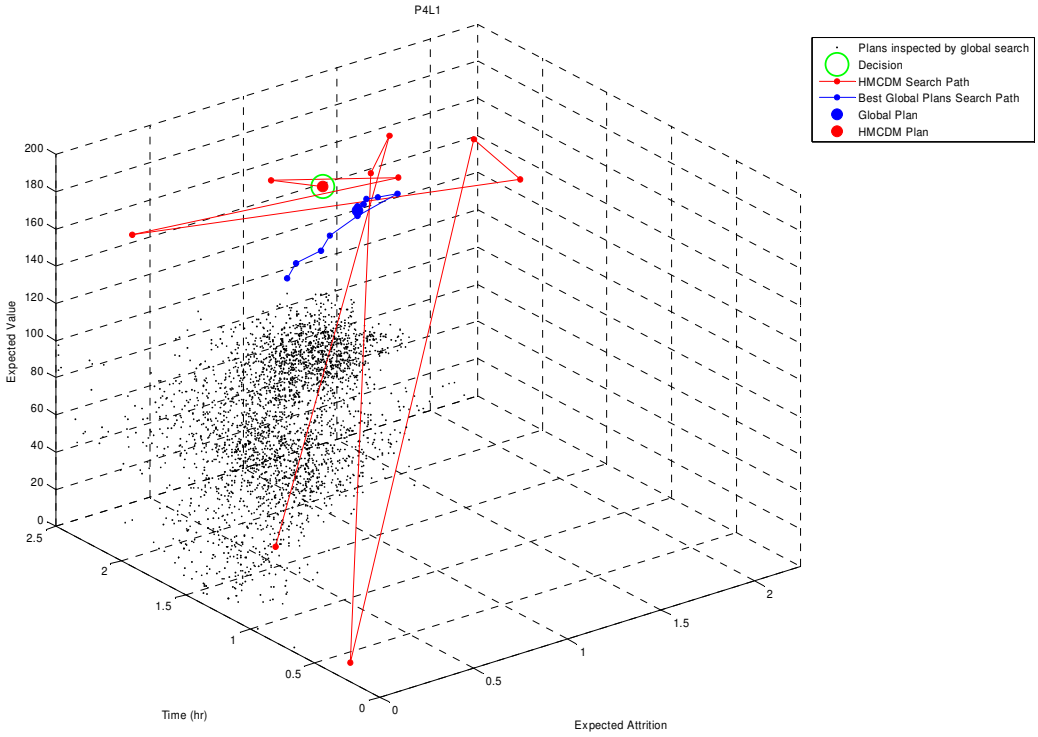

P2L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

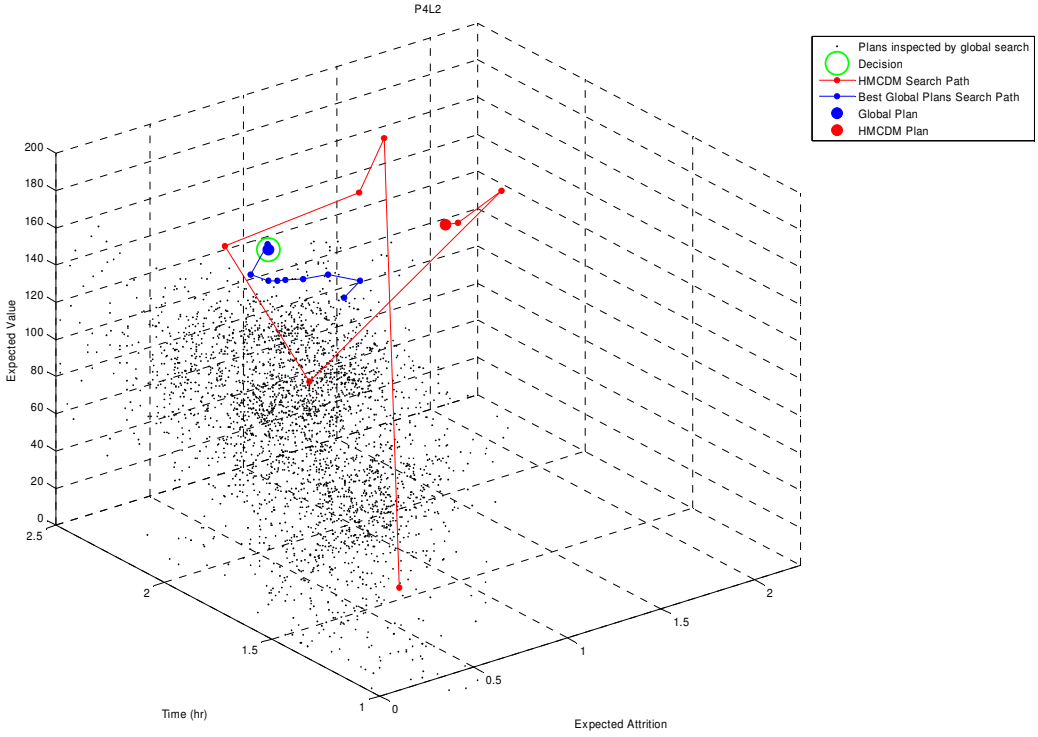

P3L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

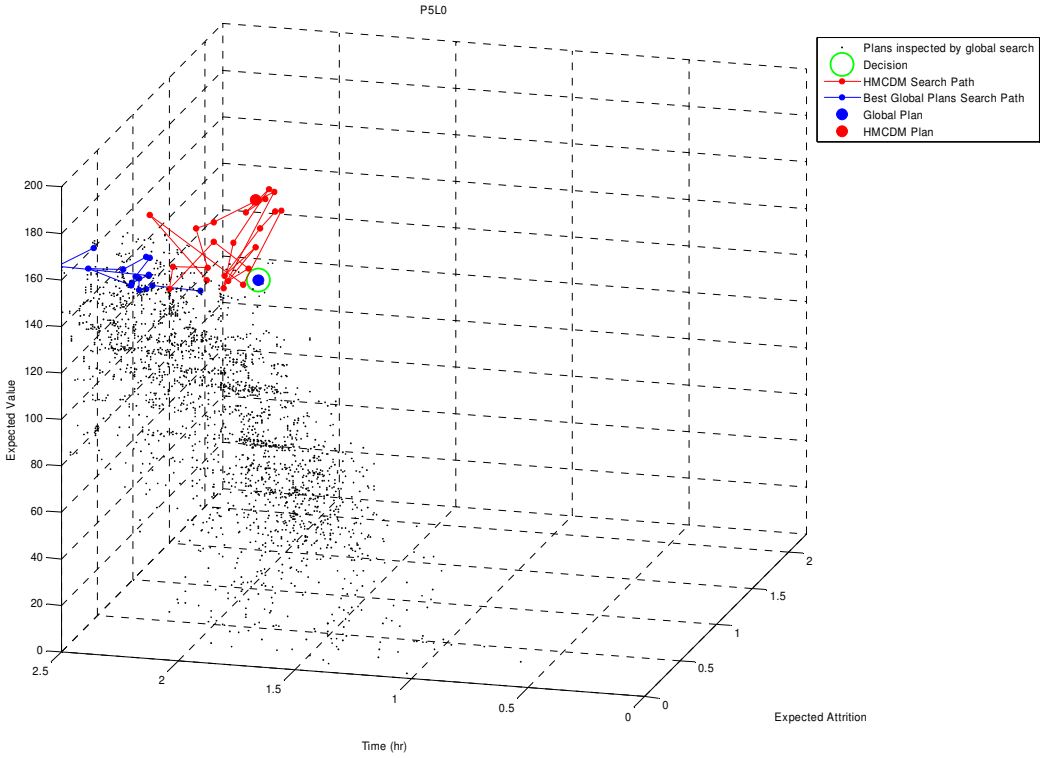

P3L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

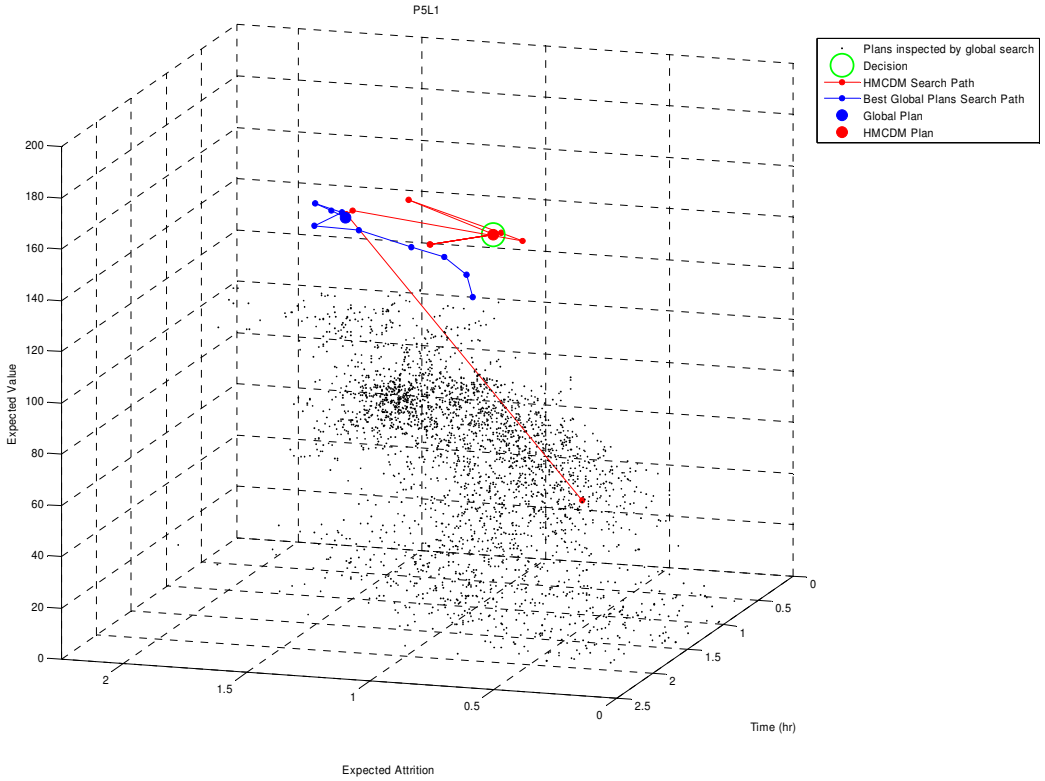

P3L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

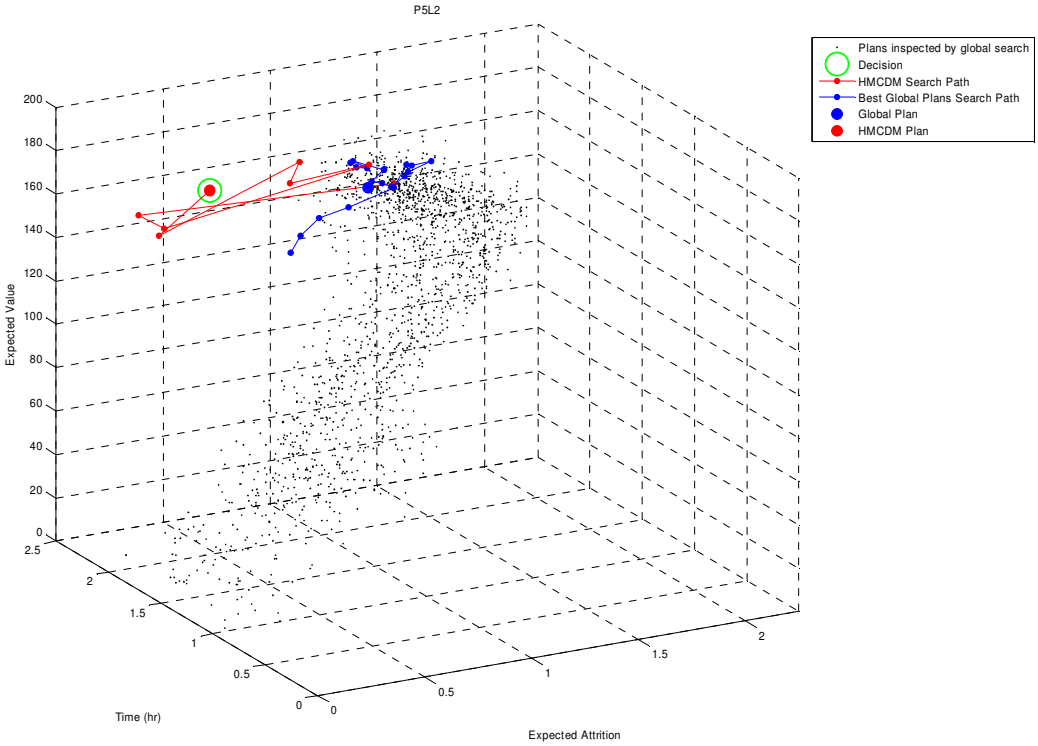

P4L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

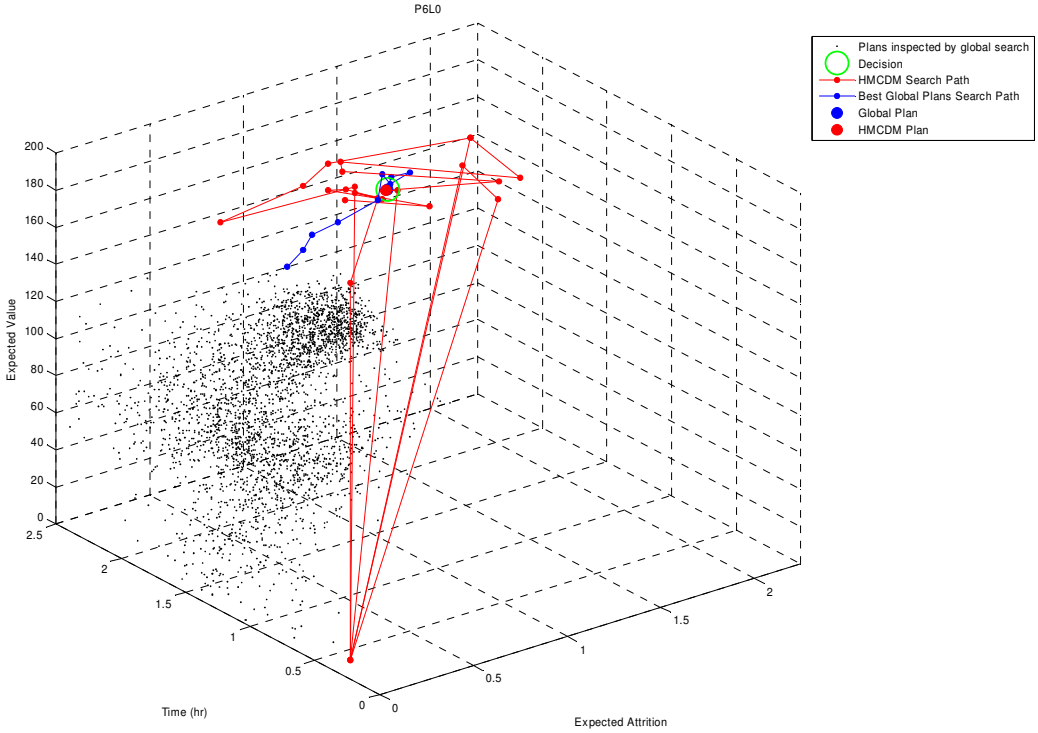

P4L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

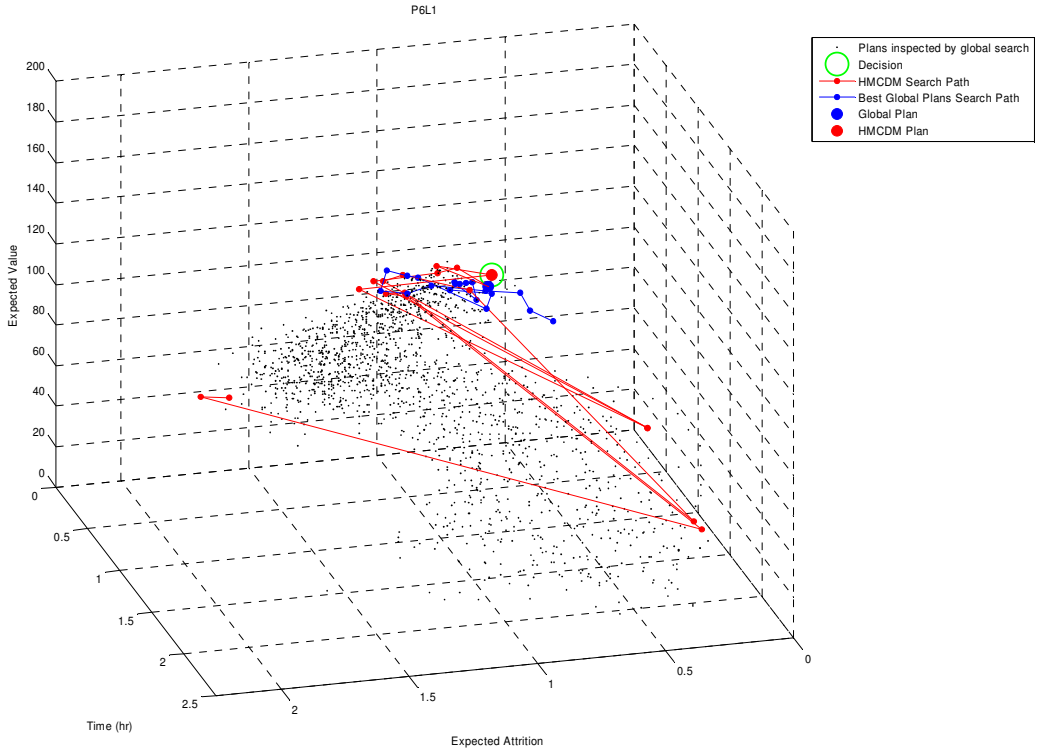

P4L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

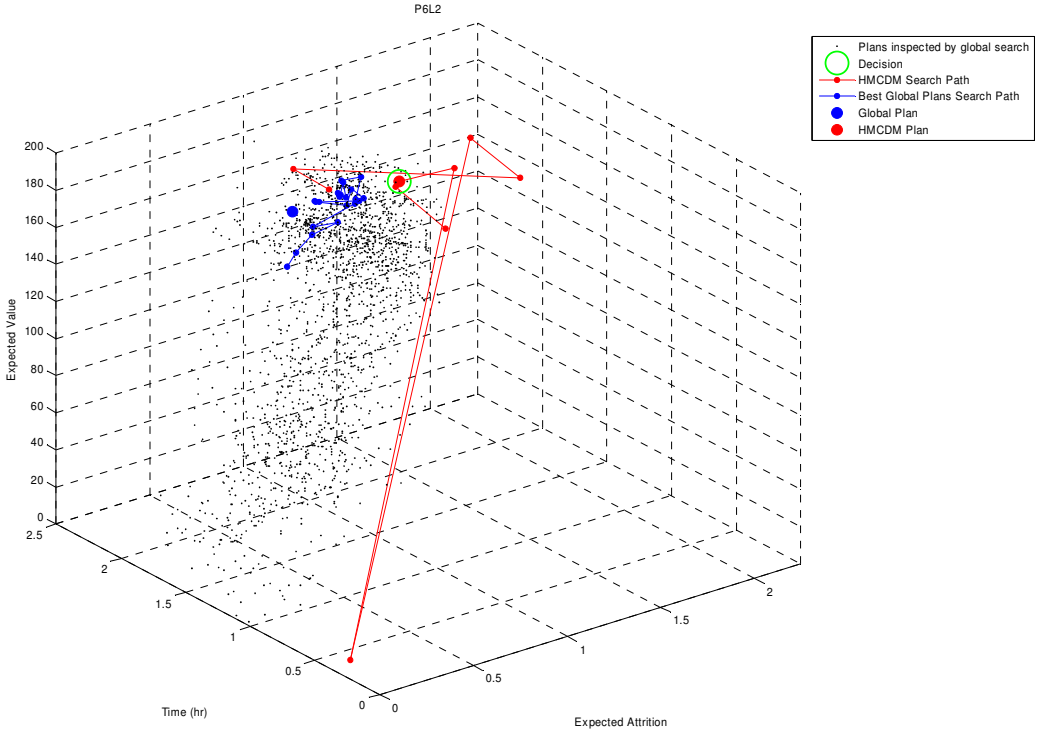

P5L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

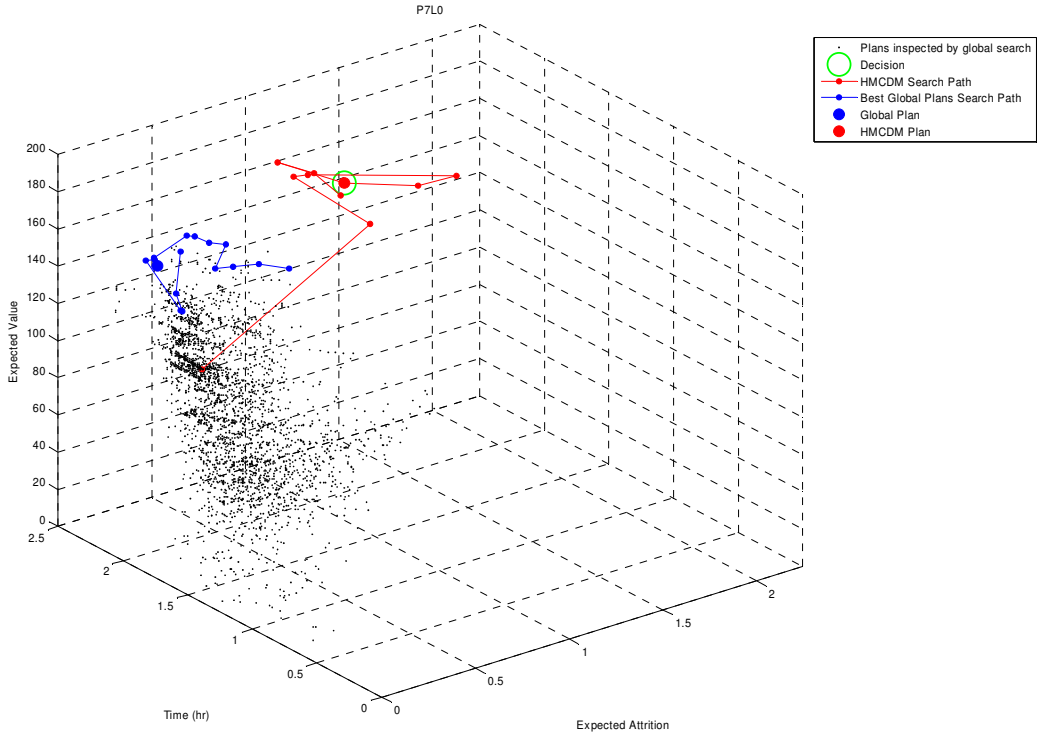

P5L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

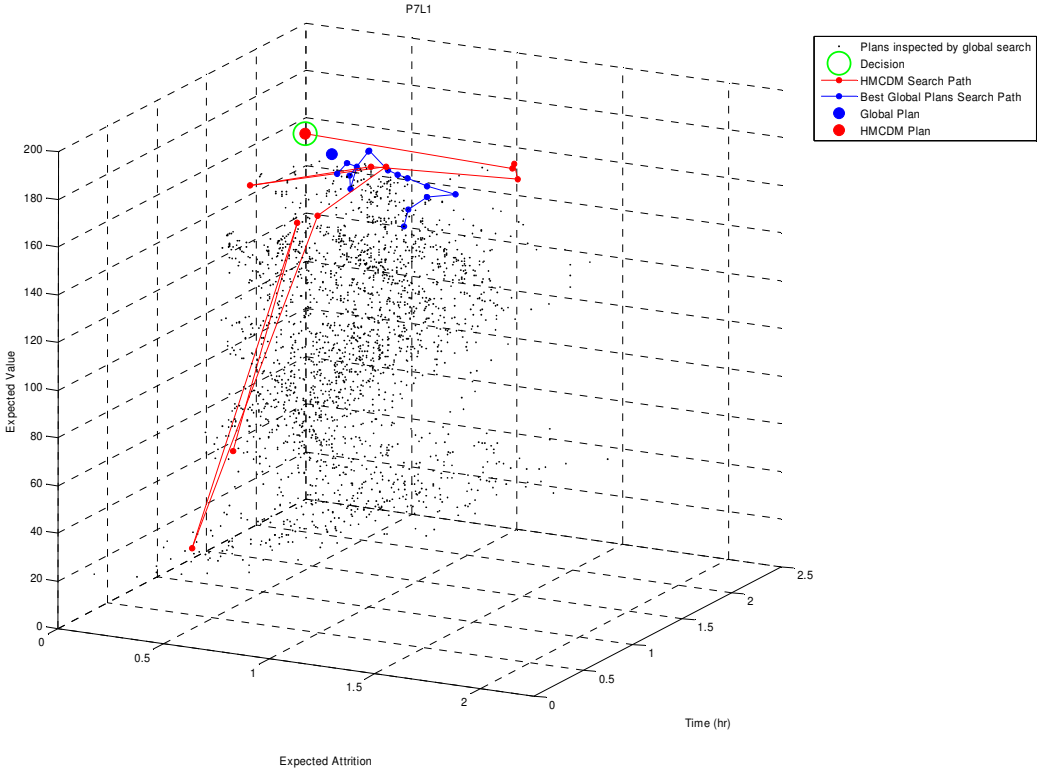

P5L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

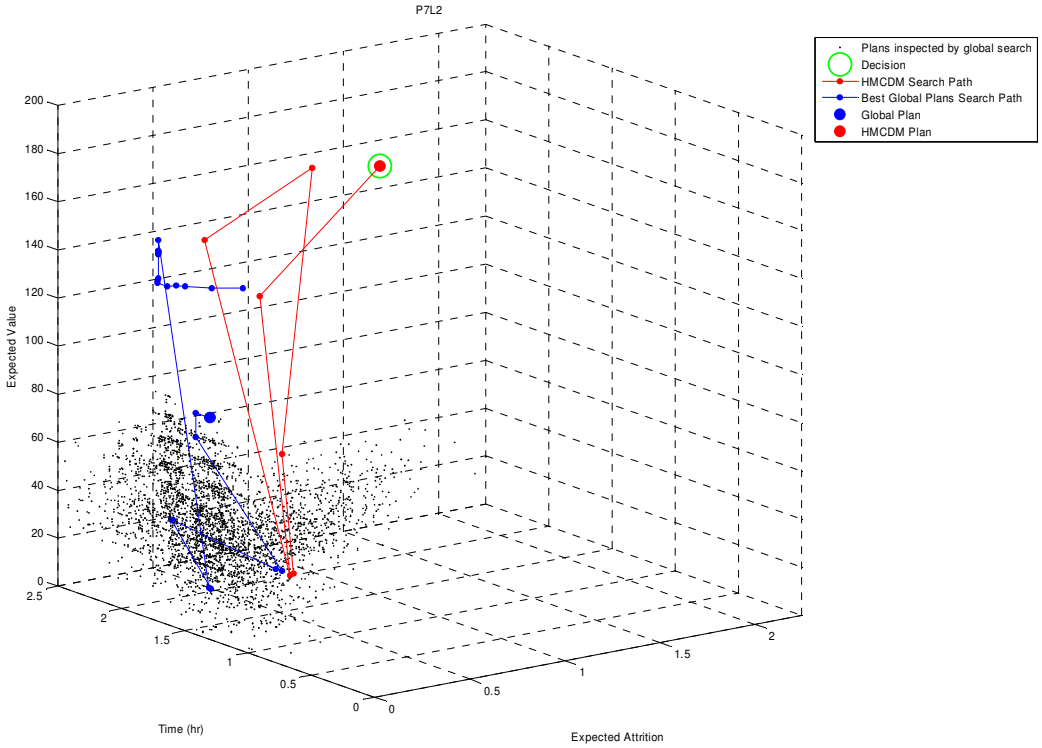

P6L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

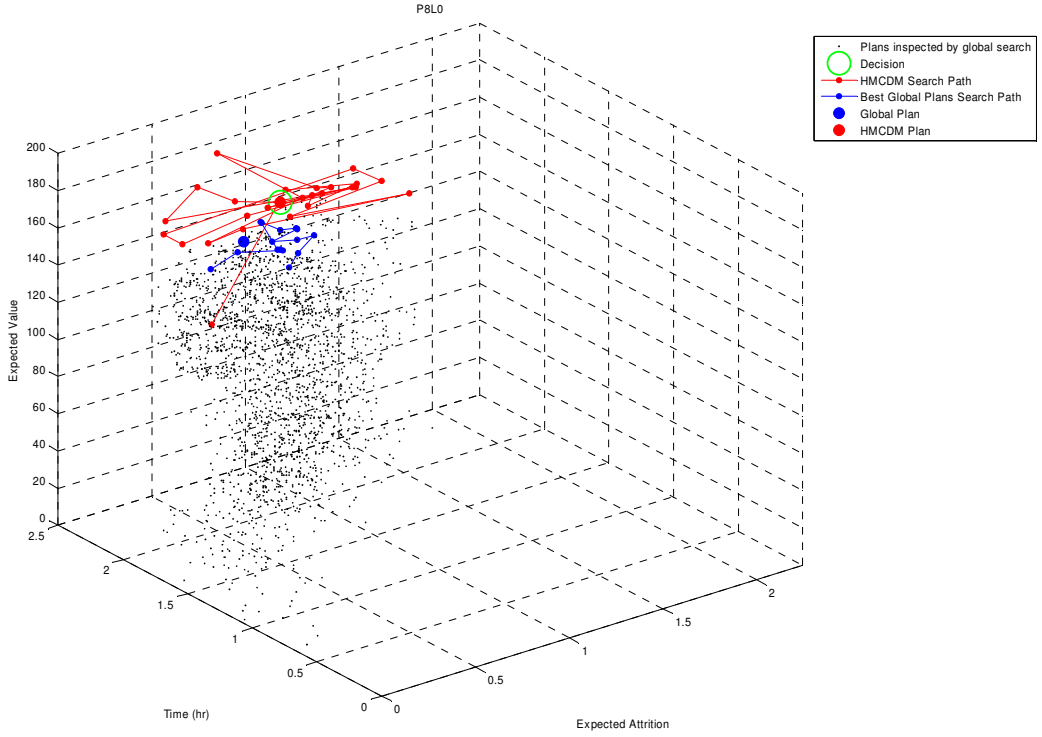

P6L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

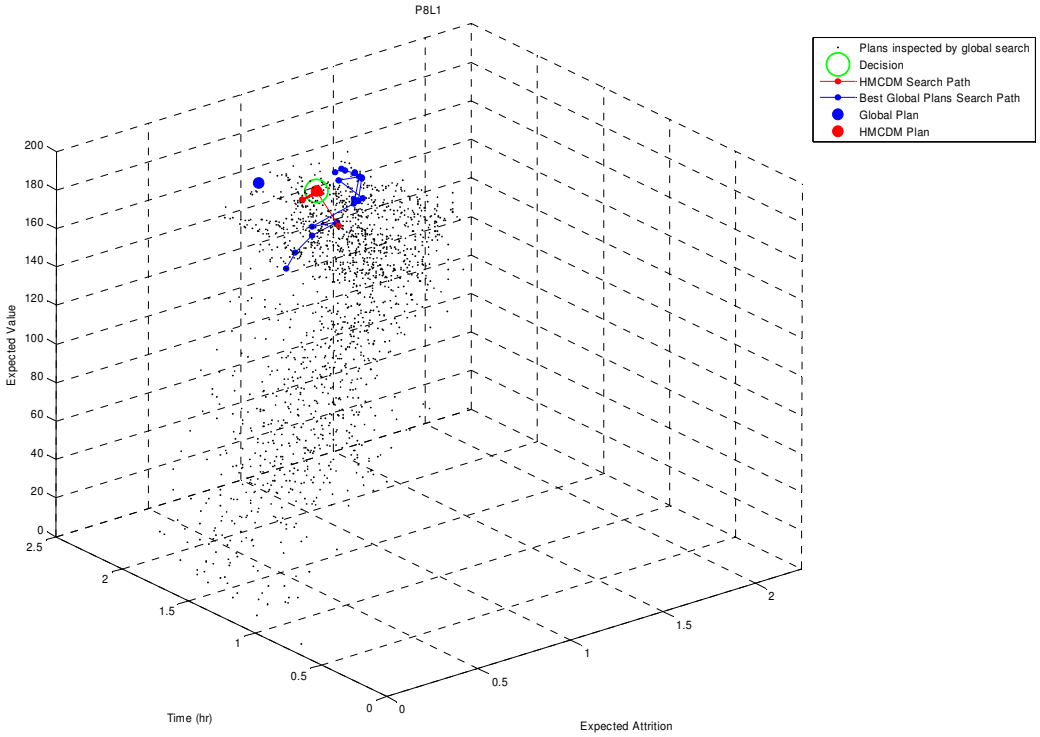

P6L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

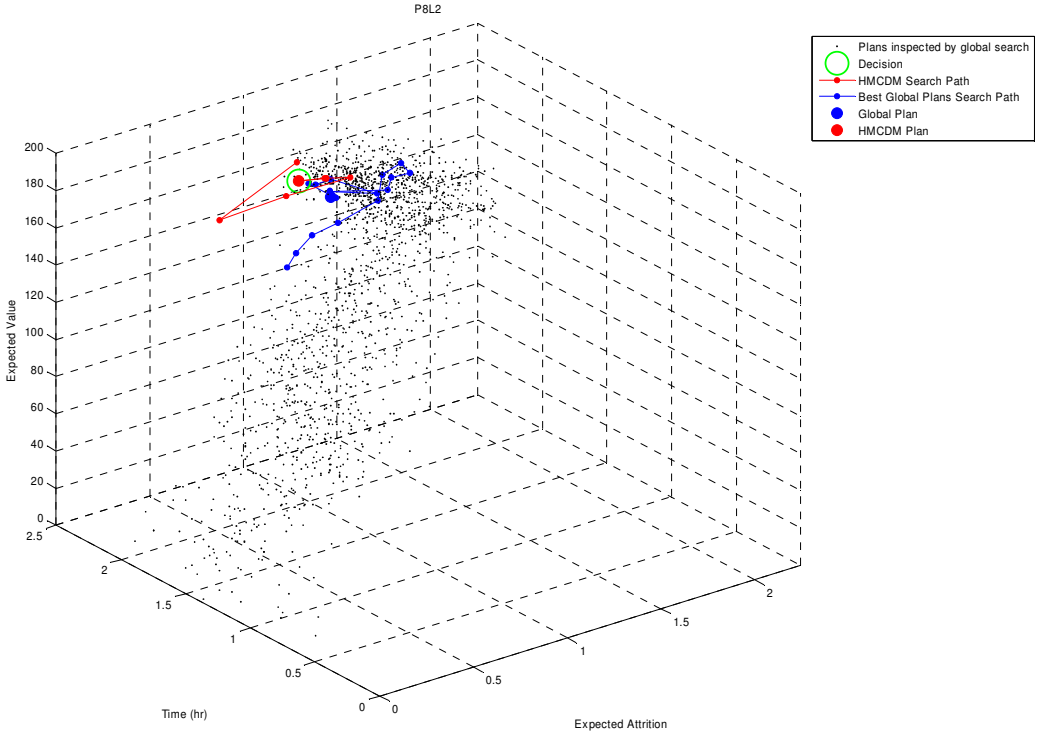

P7L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition


P7L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Expected Attrition
Time (hr)


P7L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition


P8L0
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition


P8L1
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition


P8L2
Plans inspected by global search
Decision
HMCDM Search Path
Best Global Plans Search Path
Global Plan
HMCDM Plan
Expected Value
Time (hr)
Expected Attrition

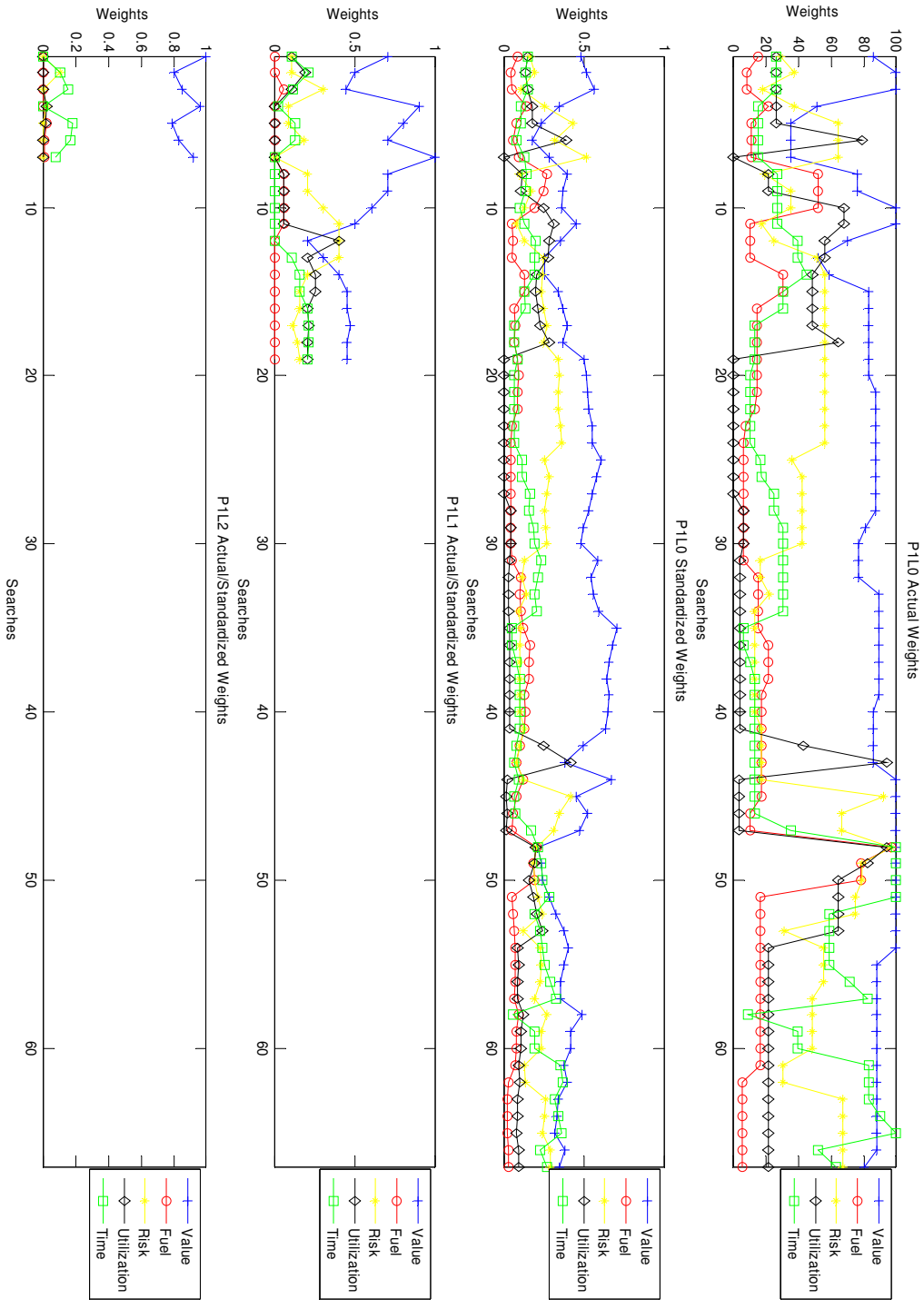
[This Page Intentionally Left Blank]

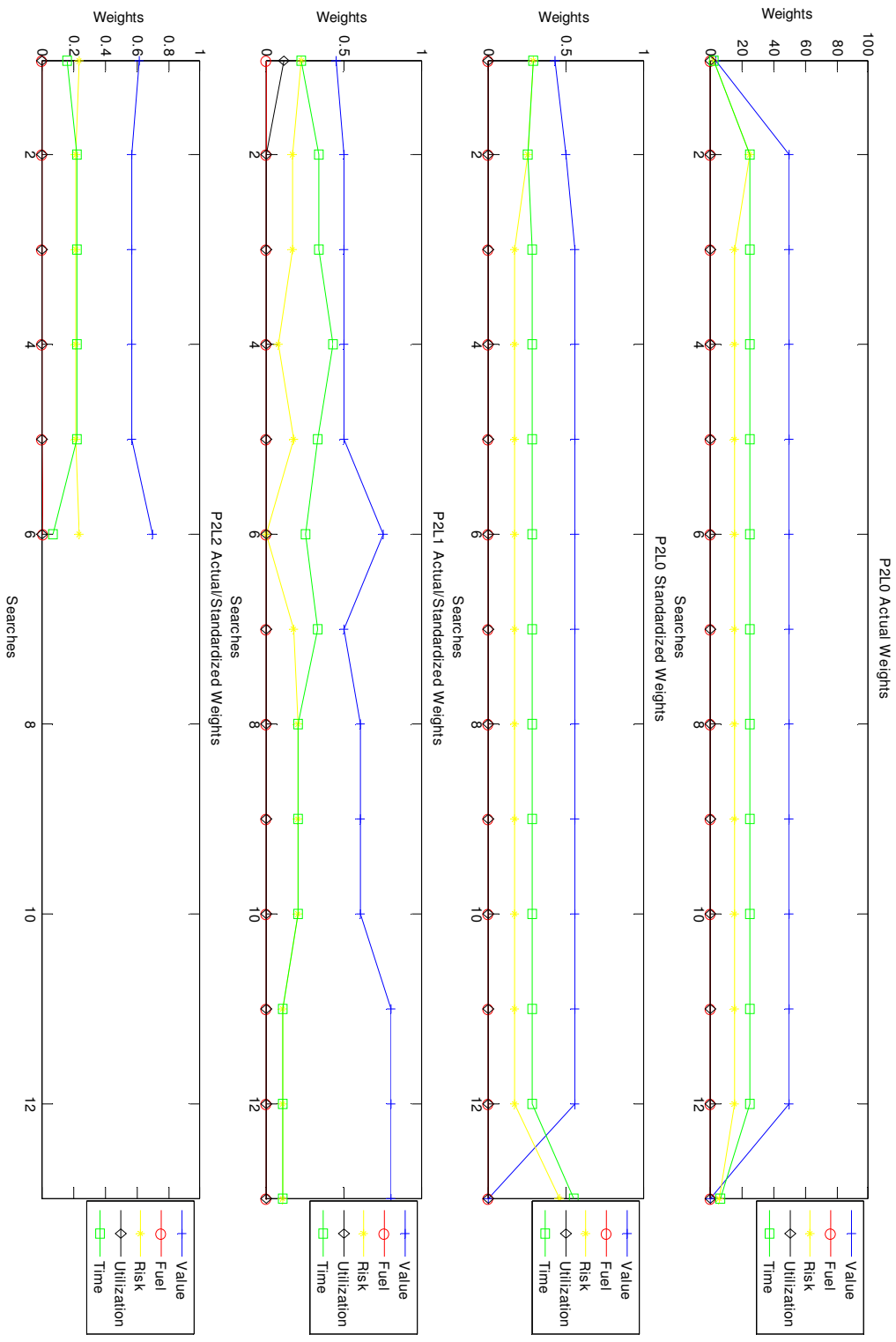
Appendix D

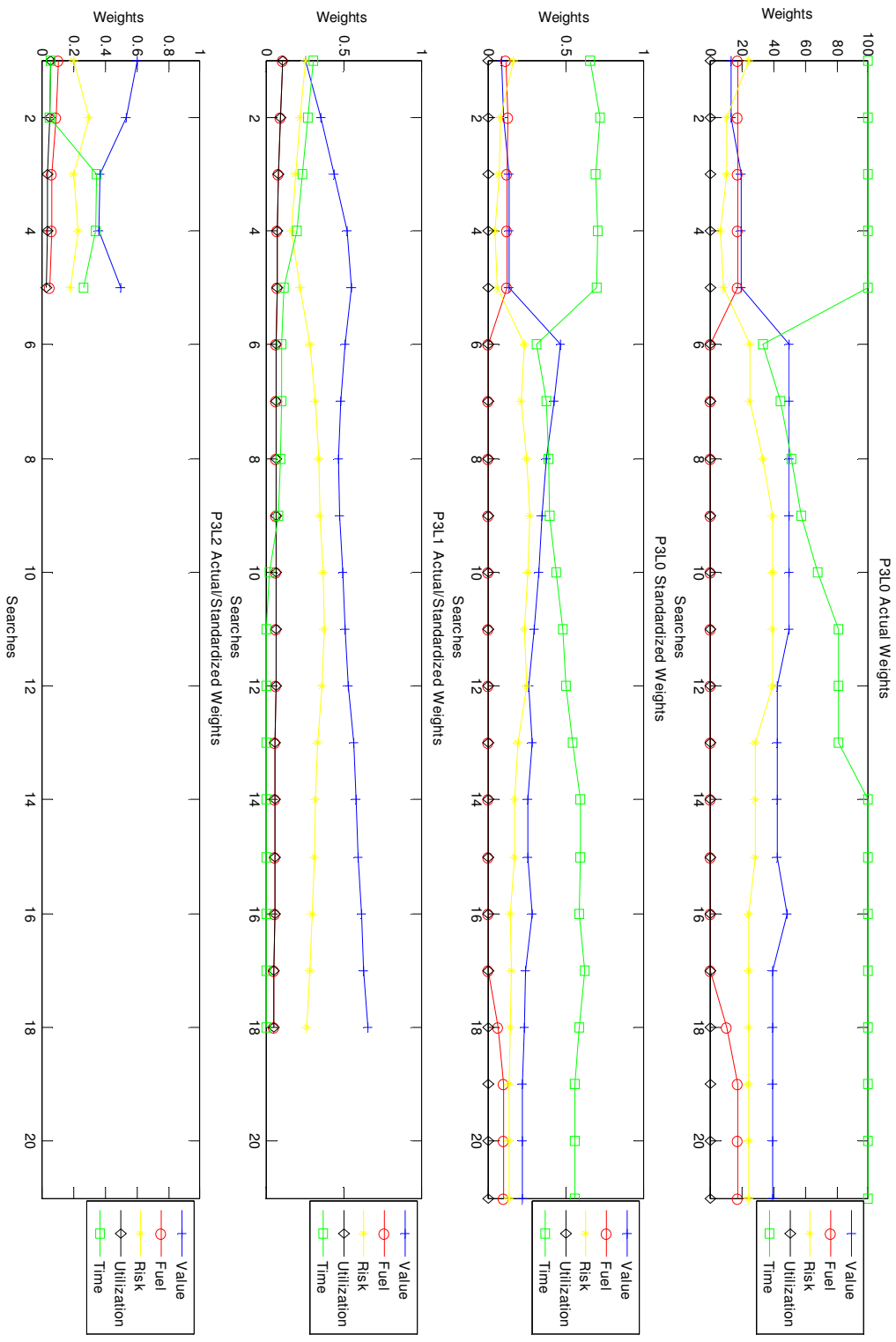
Coefficient Plots

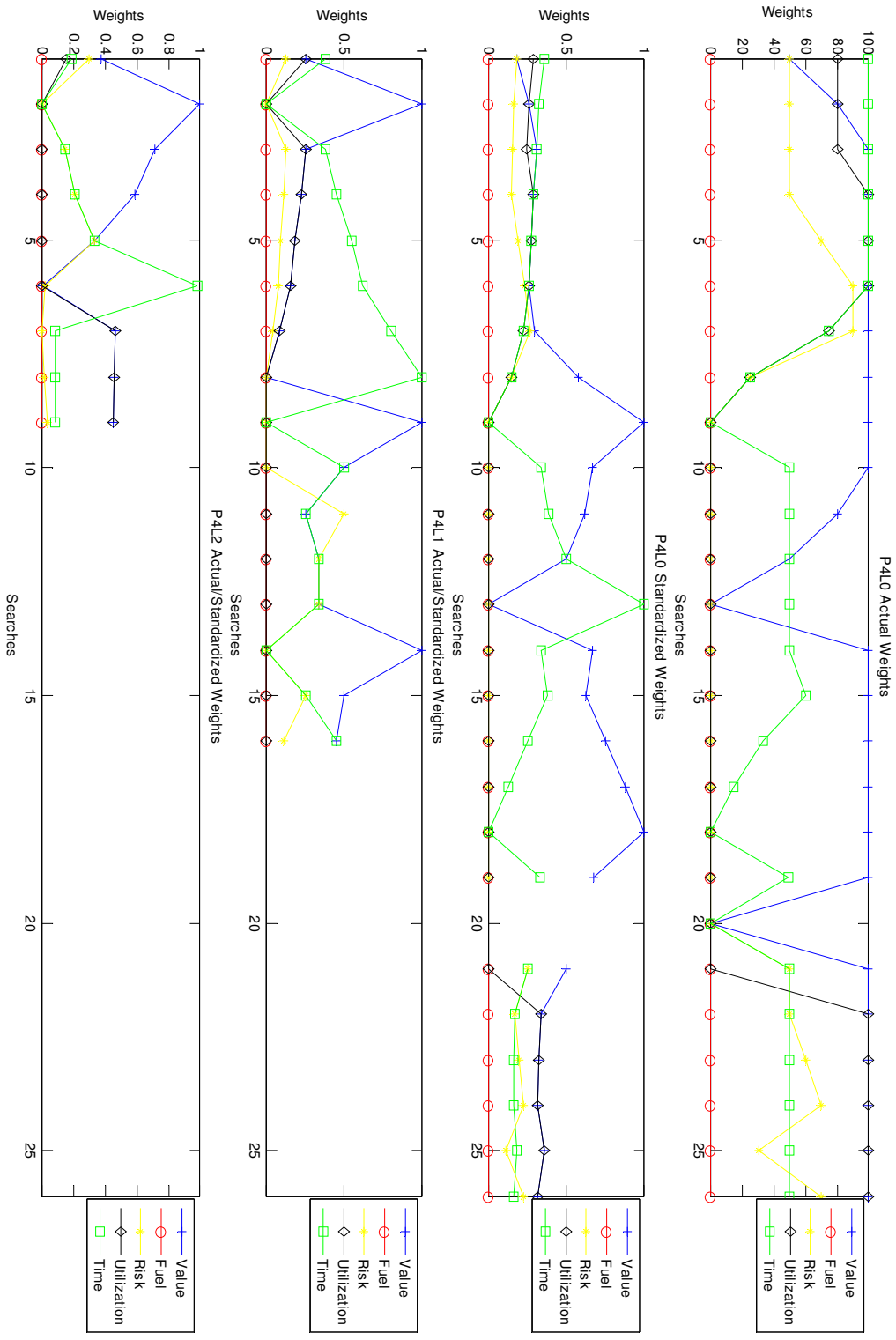
MATLAB was used to plot the weights used for each HGA search. Each plot contains four subplots that can be used to compare how the user selected coefficients across experimental levels. The x-axis is the number of searches performed. The y-axis is the weights. The first subplot shows the weights of the Level 0 search and its y-axis is scaled from 0 to 100, like the Sliding Bars. The second subplot standardizes the weights of the Level 0 search so that they sum to 1. This allows for a comparison between the actual weights on the first subplot and their relative values in subplot 2, as would be shown on a pie chart. The third and fourth subplots respectively show the standardized weights for Levels 1 and 2, as would be shown on the pie chart. The legend explains which colors and shapes represent each weight.

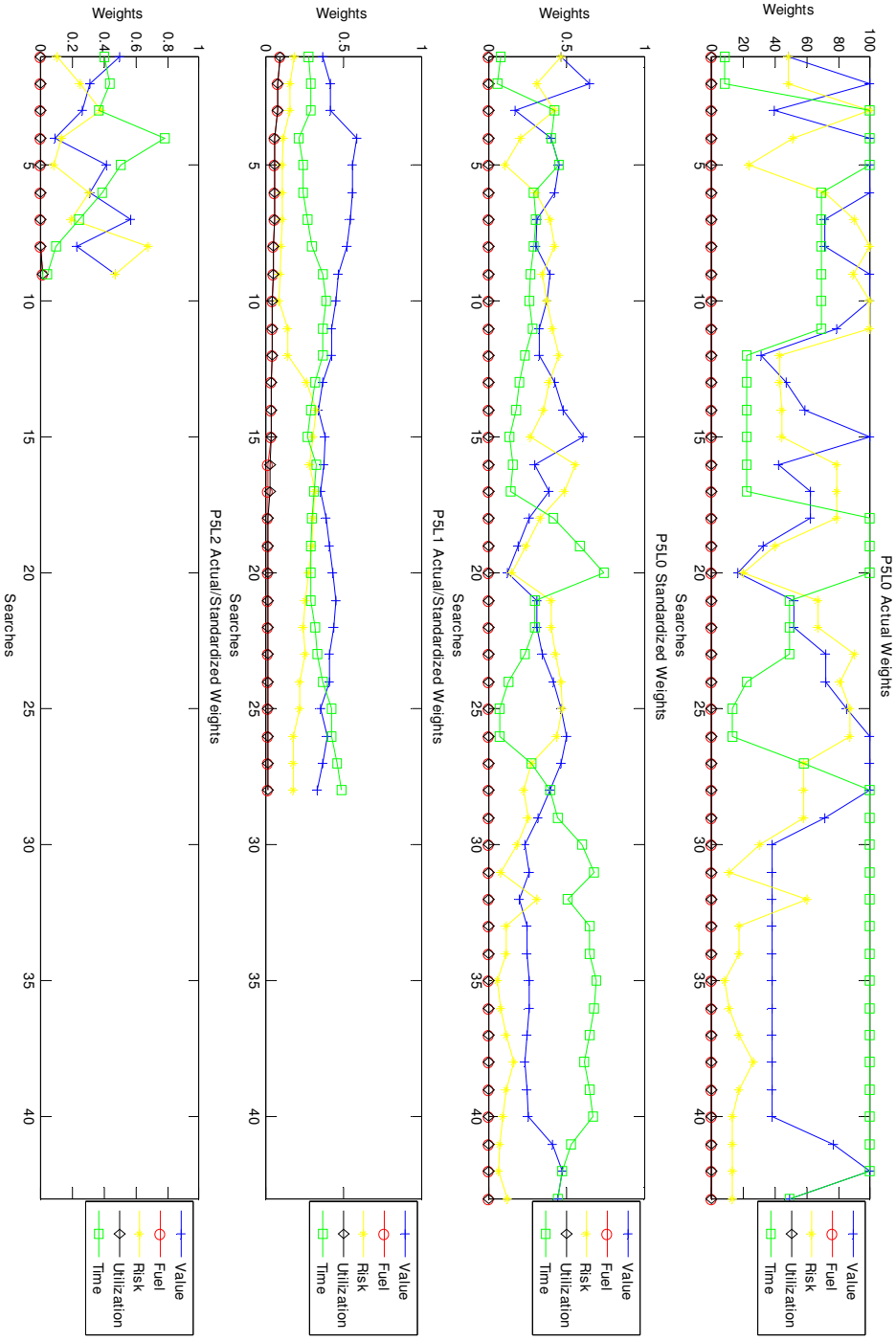
The plots are labeled P*X*L*Y,* meaning participant *X* level *Y*. For example, P7L2 is participant 7 Level 2 of TBD.

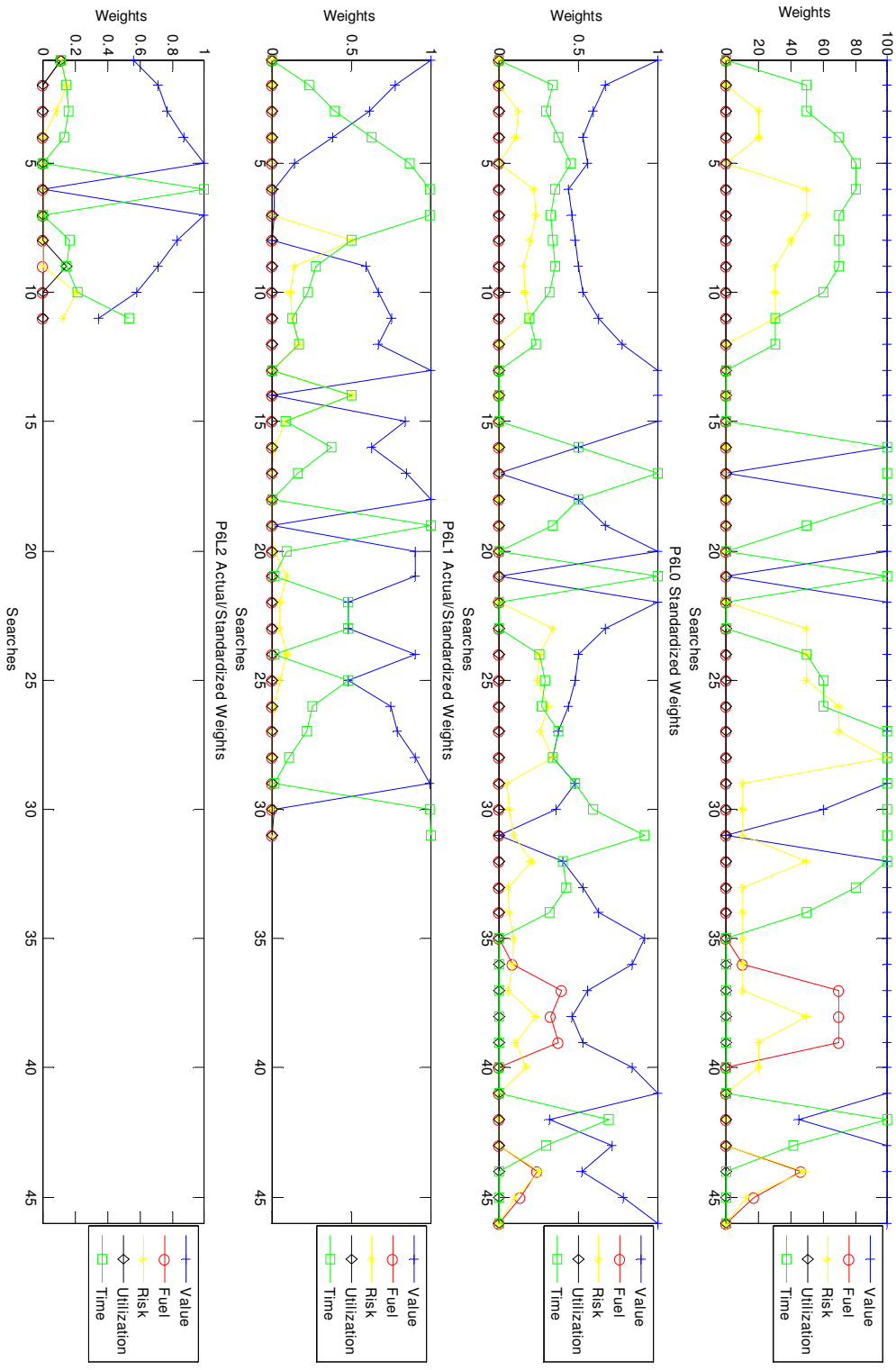

P1L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P1L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P1L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P1L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time

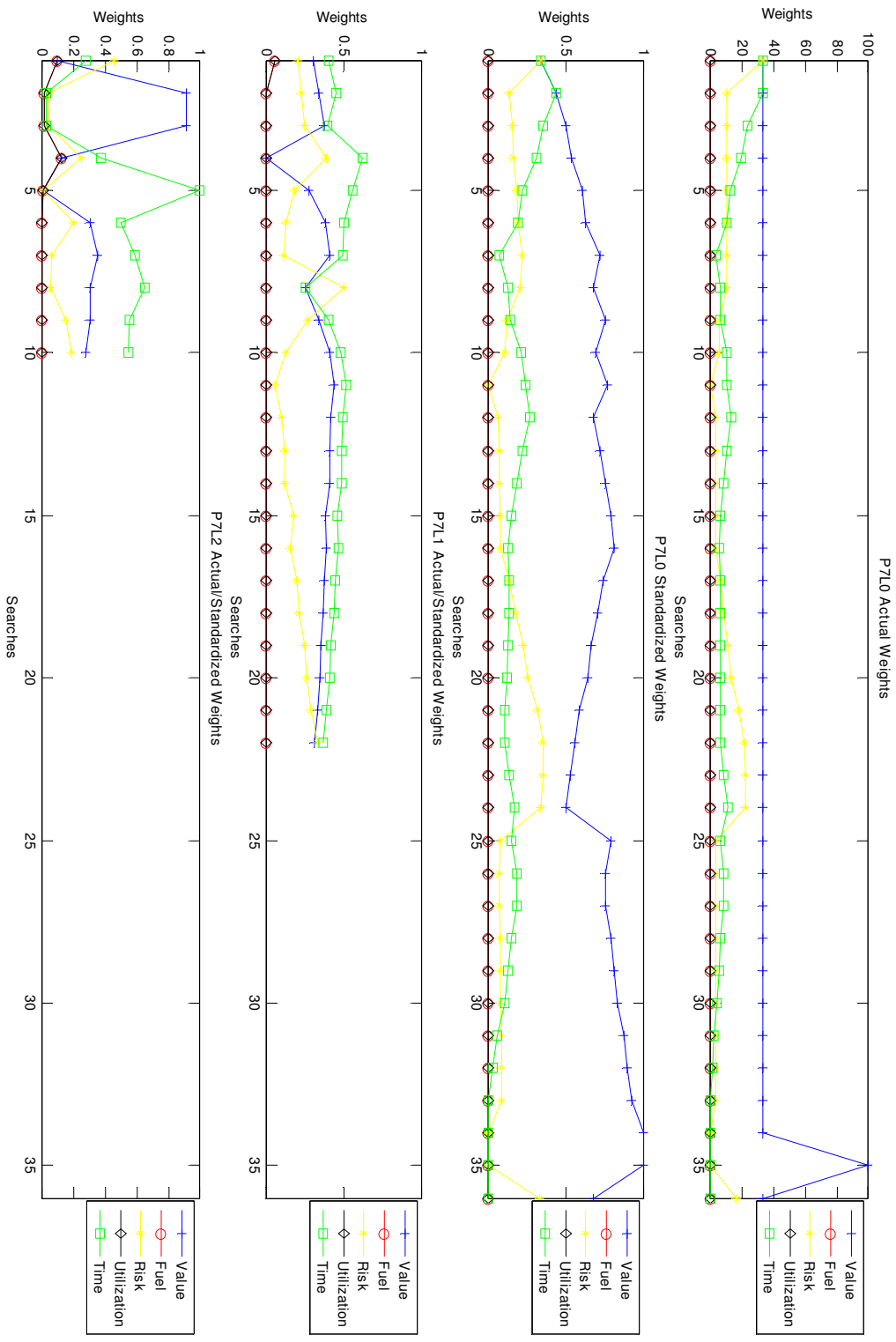

P2L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P2L0 Standardized Weights
P2L1 Actual/Standardized Weights
P2L2 Actual/Standardized Weights

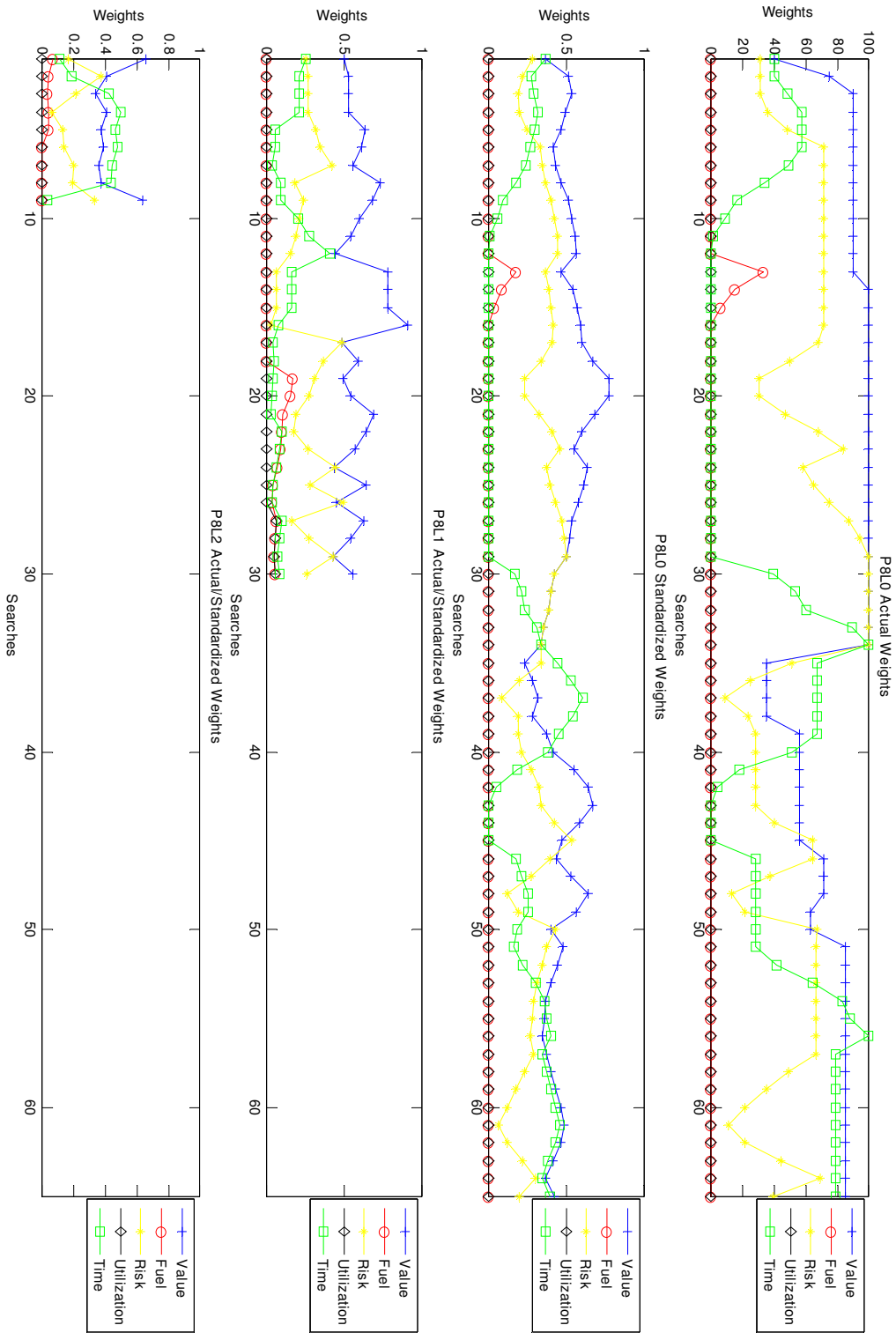

P3L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P3L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P3L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P3L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time


P4L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P4L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P4L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P4L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time


P5L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P5L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P5L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P5L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time


P6L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P6L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P6L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P6L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time


P7L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P7L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P7L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P7L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time


P8L0 Actual Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P8L0 Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P8L1 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time
P8L2 Actual/Standardized Weights
Weights
Searches
Value
Fuel
Risk
Utilization
Time

[This Page Intentionally Left Blank]

Table of Acronyms

ANOVA: Analysis of Variance
APT: Argument-based Probabilistic Trust
ARMAV: Autoregressive Moving Average Vector
CVRTW: Capacitated Vehicle Routing problem with Time Windows
DSV: Decision Space Visualization tool, Draper Laboratory
DOD: Department of Defense
EID: Ecological Interface Design
GUI: Graphical User Interface
HCC: Human Complimentary Approach
HGA: Human-Guided Algorithm
HMCDM: Human Machine Collaborative Decision Making
HuGS: Human Guided Search tool, MERL
LOC1-LOC4: Level of Collaboration 1 – Level of Collaboration 4
MERL: Mitsubishi Electron Research Laboratories
MICA: Mixed-Initiative Control of Automa-teams
MIT: Massachusetts Institute of Technology
OODA: Observe, Orient, Decide, and Act
P1-P8: Participant 1 – Participant 8
SAM: Surface to Air Missile
TBD: Trust-Based Design
TSP: Traveling Salesman Problem
UAV: Unmanned Aerial Vehicle
USAF: United States Air Force
USAFA: United States Air Force Academy

[This Page Intentionally Left Blank]

Bibliography

[1] Cohen, M., Parasuraman, R., & Freeman, J. 1998, 'Trust in Decision Aids: What Is It and How Can It Be Improved?', Proceedings *of the Command and Control Research & Technology Symposium, Monterey*, CA.

[2] Boyd, J. R. 1996, 'The Essence of Winning and Losing', http://www.belisarius.com/modern_business_strategy/boyd/essence/eowl_frameset.htm.

[3] DeGregory, Keith. 2007, 'Optimization-Based Allocation of Force Protection Resources in an Asymmetric Environment', MIT.

[4] Dzindolet, et al. 2003, 'The Role of Trust in Automation Reliance', *International Journal of Human-Computer Studies Archive.* Spec. issue: Trust and Technology, vol. 58, pp. 697-718.

[5] Forest, Laura, et al. 2007, 'The Design and Evaluation of Human-Guided Algorithms for Mission Planning', *Proceedings of the 2007 Human Systems Integration Symposium*, Annapolis, MA.

[6] Fitt, P.M. 1951, 'Human Engineering for an Effective Air Navigation and Traffic Control System', Washington D.C.: National Research Council. 1951.

[7] Gao, Ji, & Lee, John. 2005, 'Extending the Decision Field Theory to Model Operators' Reliance on Automation in Supervisory Control Situations', *IEEE Transactions on Systems, Man, and Cybernetics.*

[8] Itoh, M, Abe G, & Tanaka K. 1999, 'Trust in and Use of Automation: Their Dependence on Occurrence Patterns of Malfunctions', *IEEE.*

[9] Johnson, Andrea. 2007, 'Optimal Estimation of Ionosphere-Induced Group Delays of Global Positioning Satellite Signals During Launch, Orbit and Re-Entry', MIT.

[10] Klau, et al. 2002, 'Human Guided Tabu Search', *American Association For Artificial Intelligence,* MERL.

[11] Klau, et al. 2002, 'The HuGS Platform: A Toolkit for Interactive Optimization', *Advanced Visual Interfaces,* MERL.

[12] Krenzke, Thomas. 2006, 'Ant Colony Optimization for Agile Motion Planning', MIT.

[13] Lee, John, & Moray, Neville. 1992, 'Trust, Control Strategies and Allocation of Function in Human Machine Systems', *Ergonomics,* vol. 35, pp. 1243-70.

[14] Lee, J., & Moray N. 1994, 'Trust, Self Confidence, and Operators Adaptation to Automation', *International Journal of Human-Computer Studies*, vol. 40, pp.153-184.

[15] Lee, John, & See, Katrina. 2004, 'Trust in Automation: Designing for Appropriate Reliance', *Human Factors,* vol. 46, pp. 50-80.

[16] Liu, Cheng-Li & Hwang, Sheue-Ling. 1999, 'Application of Quality Engineering to Evaluate Effects of Situational Awareness and Trust in Automation System', *IEEE*.

[17] Malasky, Jeremy. 2005, 'Human Machine Collaborative Decision Making in a Complex Optimization System', Masters Thesis. MIT.

[18] Martin, Kiel. 2007, 'Dynamic Planning under Uncertainty for Theater Airlift Operations', MIT.

[19] Masalonis, Anthony. 2003, 'Effects of Training Operators on Situation-Specific Automation Reliability', *IEEE*.

[20] Moran, Patrick. 2006, 'OODA.gif', Jul 2006. <http://en.wikipedia.org/wiki/OODA_Loop>.

[21] Mosier, K, & Skita, L. 1996, 'Human Decision Makers and Automated Decision Aids: Made For Each Other?', In R. Parasuraman & M. Mouloua (Eds.), 'Automation and Human Performance: Theory and Applications', Hillsdale, NJ: Lawrence Erlbaum Associates, Inc. pp. 201-20.

[22] Muir B. 1987, 'Trust between humans and machines, and the design of decision aids', *International Journal of Man-machine Studies*, vol. 27, pp. 527-539.

[23] Muir, B. 1994, 'Trust in Automation: Part I: Theoretical Issues in the Study of Trust and Human Intervention in Automated Systems', *Ergonomics*, vol. 37, pp. 1905-22.

[24] Muir B.M. & Moray N.P. 1996, 'Trust in Automation: Part II: Experimental Studies of Trust and Human Intervention in a Process Control Simulation', *Ergonomics*. vol. 39, no. 3, pp. 429-460.

[25] Parasuraman R., & Riley V. 1997, 'Humans and Automation: Use, Misuse, Disuse, and Abuse', *Human Factors.* vol. 39, pp. 230-252.

[26] Parasuraman, Sheridan, & Wickens. 2000, 'A Model for Types and Levels of Human Interaction with Automation', *IEEE Transactions on Systems, Man, and Cybernetics.* 2000.

[27] Prabhala, & Gallimore. 2004, 'Investigation of Error Rates When Controlling Multiple Uninhabited Combat Aerial Vehicles', *Winter Simulation Conference*.

[28] Prabhala, Gallimore, & Narayanan. 2003, 'Human Effectiveness Issues in Simulated Uninhabited Combat Aerial Vehicles', *Winter Simulation Conference*.

[29] Rabiej, Dominik. 2000, 'Evaluating and Improving Human-Guided Simple Search with Heuristics', *Siemens Westinghouse Science and Technology Competition.* MERL 2000.

[30] Rasmussen, Jens. 1999, 'Ecological Interface Design for Reliable Human-Machine Systems', *International Journal of Aviation Psychology,* vol. 9, pp. 203-223.

[31] Ruff A.H, Narayanan S, & Draper, M. 2002, 'Human Interaction with Levels of Automation and Decision-Aid Fidelity in the Supervisory Control of Multiple Simulated Unmanned Air vehicles', *Presence,* MIT, vol. 11, pp. 335-351.

[32] Schmitt, J., & Klein, G. 1999, 'A Recognitional Planning Model', *Proceedings of the 1999 Command and Control Research and Technology Symposium,* Newport, Rhode Island: Naval War College.

[33] Scott, et al. 2002, 'Investigating Human-Computer Optimization', *CHI.* MERL.

[34] Shahroudi, K.E. 1997, 'Design by Collaboration between Manual and Automatic Optimization', *Stichting Mathematisch Centrum: CWI.*

[35] Sheridan, Thomas B. 2002, *Humans and Automation: System Design and Research Issues,* Santa Monica, CA: Wiley Pub.

[36] Skitka, Mosier, & Burdick. 1999, 'Does Automation Bias Decision-Making?', *Int. J. Human-Computer Studies.* pp. 991-1006.

[37] Terveen, Loren. 1995, 'Overview of Human-Computer Collaboration', *Knowledge Based Systems.* vol. 8.

[38] U.S. Department of Defense. 1987, 'Human Engineering Procedures Guide', Washington D.C.: DoD-HDBK-763.

[39] Vicente, Rasmussen. 1992, 'Ecological Interface Design: Theoretical Foundations', *IEEE Transactions on Systems, Man, and Cybernetics.*

[40] Vicente, Christoffersen, & Pereklita. 1995, 'Supporting Operator Problem Solving Through Ecological Interface Design', *IEEE Transactions on Systems, Man, and Cybernetics.*

[41] Wickens, C.D. & Hollands, J.G. 2000, *Engineering Psychology and Human Performance*. Prentice-Hall, Inc.: Upper Saddle River, NJ.