ABSTRACT

Title of dissertation: CREATING AN OBJECTIVE METHODOLOGY FOR HUMAN-ROBOT TEAM CONFIGURATION SELECTION

Sharon Michelle Singer
Doctor of Philosophy, 2012

Dissertation directed by: Professor David L. Akin
Department of Aerospace Engineering

As technology has been advancing and designers have been looking to future applications, it has become increasingly evident that robotic technology can be used to supplement, augment, and improve human performance of tasks. Team members can be combined in various combinations to better utilize their capabilities and skills to create more efficient and diversified operational teams. A primary obstacle to integrating new robotic technology has been the inability to quantitatively compare overall team performance between very different team configurations without limiting the analysis to a few metrics. To-date, mission designers have arbitrarily assigned importance to mission parameters, subjectively limiting the search space. While this has been effective at evaluating individual mission plans, the arbitrary evaluation criteria has made a straightforward comparison between different research projects and ranking scales impossible. The question then becomes how to select an objective set of criteria for any given problem.

It is this final question that this research sought to answer. A methodology was developed to facilitate performance comparison amongst heterogeneous human and

robot teams. This methodology makes no assumptions about mission priorities or preferences. Instead, it provides an objective, generic, quantitative method to reduce the complexity of the mission designer's decision space. It employs an heuristic, greedy objective reduction algorithm to reduce problem complexity and a multi-objective genetic algorithm to explore the design space.

The human-robot team configuration selection problem was utilized as the application that motivated this research. The methodology, however, will be applicable to a wider domain of research. It will provide a structure to enable broader search of the design space, exploration of the differences between performance metrics, and comparison of optimization models that facilitate evaluation of the design options.

CREATING AN OBJECTIVE METHODOLOGY FOR HUMAN-ROBOT TEAM CONFIGURATION SELECTION

by

Sharon Michelle Singer

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor David L. Akin, Chair/Advisor
Professor Shapour Azarm
Professor Linda Schmidt
Professor Raymond Sedwick
Professor Norman Werely

Acknowledgments

There have been so many supportive people who have helped me along this long and windy road. Foremost, I would like to thank Dr. Dave Akin for all of his help over the years. As my academic and research advisor, he encouraged me to pursue my own vision and research interests. Dr. Akin has an astounding wealth of knowledge on everything space related and was a great resource for framing any outlandish ideas into a more concrete experimental reality.

I would like to thank all of the members of my dissertation committee for their time, support and contributions over the years. Each member has brought their own expertise to the table, creating a constructive, diverse sounding board for my research ideas.

I would like to recognize the phenomenal aid of Dr. Dimo Brockhoff. Not only does his objective reduction algorithm form the backbone of this dissertation, but he was very receptive and patient when I reached out to him for assistance in understanding a portion of his algorithm. Without his time, aid and collaboration, this dissertation would not exist. I am very thankful for all of the support he has given me.

I would also like to recognize and thank Dr. Mary Bowden for her compassionate advice, advocacy, support, and encouragement over the years. She was the first representative of the Aerospace Engineering Department that I met when I first came to the University of Maryland, and she has always been warm and welcoming, always willing to listen to my new challenges and triumphs.

To the Space Systems Laboratory at large: Through his collaborative leadership style, Dr. Akin has fostered a wonderfully supportive research environment within the Space Systems Laboratory. Students are encouraged to work together, to come up with creative solutions to problems and to continue exploring and asking "what if?". In my tenure at the lab, the SSL has drawn several generations of the most dedicated, enthusiastic, and inspiring group of graduate students. Whether serving as knowledge experts and consultants for each other, sharing their own experiences as lessons learned, or providing the levity and comic relief necessary to keep us all sane through the long hours and long years, the students of the SSL have been amazingly supportive. I could not have made it to where I am today without you all. Whether it was attending a Viking funeral for the lab's pet crawfish, competing with the other scuba divers for the most time spent underwater in a given year (the lab's "Idle Hands" award), or the retreats away from the lab and out into the real world to enjoy sunshine and good company, I thank you all.

I would especially like to recognize the phenomenal support of a few fellow graduate students. To Martin Stolen, for taking me under his wing, teaching me all he knew about human and robot interaction, and encouraging me to continue into the PhD program. To John Mularski for sharing his love of scuba diving and Barrett Dillow for convincing me to take the plunge and become certified! To Joe Galante, Barrett Dillow, and Craig Lewandowski for making me feel like a part of the lab, and for insisting that I have a life outside of it, too! To Kate McBryan for her exuberance, friendship, enthusiastic crafting, technical expertise, and for the hours she spent helping me disect an algorithm ... To Angela Peace, for aid with the aforementioned algorithm and as my math knowledge expert reference. There are so many other friends who have helped me along the way, it would take another chapter to name them all.

It goes without saying that my friends and family have been wonderfully supportive over the years. Your love, confidence, well-wishes, and belief in me have carried me through graduate school.

Finally, I would like to thank Rick Barnard, my soon-to-be husband. He has been so understanding and patient of all of my frustrations in the final months of this dissertation effort, always believing in me and my abilities. Whether available to listen to my most recent mini-problems or mini-triumphs, for insisting on hearing the long version of all of my stories, or going for great walks around the lake on weekends, he has always been there for me, with love and support for all of my endeavors.

Thank you all for your esteem and support. It has meant the world to me.

Table of Contents

List of Figures

List of Abbreviations

C4T	Controller-to-Controller Communication and Coordination Taxonomy: model of team communication (FAA's software package)
CD	coordination demand (task performance metric)
CTR	critical time ratio (task performance metric)
EVA	extra-vehicular activity
FAA	Federal Aviation Administration
FAT	false alarm time (task performance metric)
FO	fan out (task performance metric)
HRI	human robot interaction
HRI/OS	Human Robot Interaction Operating System (JPL software package)
HRT	human and robot team that cooperatively completes a mission
HST SM	Hubble Space Telescope servicing mission
HURON	Human-Robot Task Network Optimization (JPL software package)
IE	interaction effort (task performance metric)
MaOO	many-objective optimization: more than three objective functions
MITPAS	Mixed Initiative Team Performance Assessment System (Perceptronics, Inc. software package, US Army)
MOGA	multi-objective genetic algorithm
MTBI	mean time between interventions (task performance metric)
MTCI	mean time completing an intervention (task performance metric)
NASA-TLX	model to provide workload values for human mental tasks (NASA software package)
NT	neglect time (task performance metric)
OT	occupied time: differentiates between CD and demands of interacting with several independent robots (task performance metric)
RAD	robot attention demand (task performance metric)
RASCAR-CS	Robot-Assisted Search and Rescue Coding System: urban search and rescue HRI performance model (University of South Florida software package)
RMS	remote manipulator system (Space Shuttle)
RPP	robot performance parameter (task performance metric)
SC	situation coverage: percentage of total tasks in which a robot understands the directive (task performance metric)
UAV	unmanned aerial vehicle
WEI	work efficiency index (task performance metric)

Chapter 1

Introduction

1.1 Background

As technology has been advancing and designers have been looking to future applications, it has become increasingly evident that robotic technology can be used to supplement, augment, and improve human performance of tasks. With more options developing for how robotic technology can be used in an integrated human and robot team (HRT), new methods to assess individual agent performance and overall team performance have been developed.

Whether for on-orbit servicing missions or planetary exploration and habitat building, employing a cooperative human and robotic crew in future space missions will enable a larger volume of tasks to be completed. Team members can be combined in various combinations to better utilize their capabilities and skills to create more efficient and diversified operational teams.

At the present, robotic technologies have been included in space operations to aid in performing tasks that would be difficult for humans to perform. The Space Shuttle's remote manipulator system (RMS) was used to grapple with and maneuver large pieces of equipment (including the Hubble Space Telescope) and as a mobile platform to anchor astronauts. Dextre, the newest addition to the International Space Station's robotic system, is a much smaller robotic agent that provides similar support with the added ability to perform a small set of manipulative activities. Robonaut was specifically designed to perform extra-vehicular activity (EVA) tasks and to serve as a cooperative team agent with a human crew [73, 36]. Several generations of Mars rovers have demonstrated success at employing robotic arms for instrument-driven operational tasks in an environment currently uninhabitable by humans.

While robotic technologies have furthered human knowledge of and exploration of space, use of robotics in space has been limited to a few operational scenarios to replace the activities of the human crew. Interaction between these robotic technologies and the human crew has previously relegated the robots to the role of supplemental tools (as in the case of the RMS as a maneuvering platform).

There have been several investigations into the utility of human and robot partnerships for space activities. The Ranger Telerobotic Shuttle Experiment was developed to test the ability of an on-orbit dexterous robot to perform servicing activities [67]. Analysis has been performed to assess what roles in team collaboration robotic technology would be best suited to fill [3, 68]. Different ways to combine robotic technologies and hardware to aid in the human crew's productivity have been examined [2].

The diversity of robotic technology available creates a multitude of new opportunities in task performance. Utilizing the new capabilities for hardware, software, sensors and system integration, and communication architecture could lead to a greater level of mission diversity, and facilitate scenarios that had previously been impossible.

Future mission designers will have a large heterogeneous group of distinct agents (both human and robotic) from which to select the most productive or efficient team members. How can a designer select the most effective agents to complete a series of tasks? An overall team performance analysis would be beneficial and would enable quantitative comparison between disparate teams.

It is this final question that this research sought to answer. A methodology was developed to facilitate performance comparison amongst heterogeneous robot teams. This methodology makes no assumptions about mission priorities, preferences, nor importance of performance criteria. Instead, it provides an objective, generic, quantitative method to reduce the complexity of the mission designer's decision space.

In section 1.2, the problem domain addressed by this research effort will be laid out. Three distinct domains will be identified to guide the research. Each will be briefly outlined in section 1.2, and discussed in much more depth in the Literature Review of chapter 2. After discussing the state of the field, the third chapter will discuss the methodology used, and detail my unique contribution to this problem domain. Chapters 4, 5, and 6 present the test problems and applications used to demonstrate the utility of the methodology presented in this dissertation, and chapter 7 presents a discussion of the results and implications, and offers concluding remarks and suggestions for future work.

1.2 Statement of the Problem

The problem of selecting the best set of team members for a given task scenario can be divided into three orthogonal areas of research. The first area of research has been in defining and developing new technology to enable diverse human and robot teams (HRT configurations). Additionally, there has been significant research on methods to measure and quantify differences between each agents' performance in various aspects of task completion (performance metrics). Third, there has been work to optimize various parts of mission design and team selection. The research described in this dissertation will create synthesis between these three areas. It will develop a methodology that will facilitate comparison between distinct groups of team members, and will facilitate selection an optimal team for a given task scenario.

1.2.1 Differentiating HRT Configurations

Substantial research has been done to develop new technology that increases the variety of tasks and the precision with which robots can perform them. Integrating these technologies into human agent activities has the potential to increase the diversity of possible HRT configurations. As the variety increases, it becomes even more necessary to concretely define the differences between teams and to specify the performance of each participating agent.

The primary configuration differences between HRTs can be grouped into several broad categories. Whether the team's planning and performance analysis occurs offline or in real-time greatly influences the parameters to be monitored to capture the important team performance characteristics. The scenario perspective of the mission planners can determine whether the humans or robots are favored to reduce workload. For real-time mission performance, the decision and control authority structure determines where critical decisions and work allocation is made. The type of communication network and the information it is capable of passing also characterizes a HRT. The autonomy level of each robotic agent determines the amount of human intervention potentially needed in robotic tasks. How each agent (human and robot) receives information about their surrounding environment can vary greatly between different configurations. Each of these categories are described in more detail in Chapter 3.

1.2.2 Challenges to Comparing HRT Configurations - Performance Metrics

A primary obstacle to integrating new robotic technology into familiar mission scenarios is the inability to quantitatively compare overall team performance between very different team configurations. Without limiting the analysis to a few important metrics, is it possible to objectively and conclusively demonstrate whether a standard two-human International Space Station EVA team, a combined human and semi-autonomous robot team, or an autonomous robot performs an entire mission scenario better than the other configurations? Can the benefit derived from using one team configuration over another be quantified? As a designer, is it important to retain knowledge of the tasks completed by each of the agents in a given scenario separately? Or is the interest primarily in the overall team's performance of a scenario? These are the questions that drive the development of an overall team performance assessment.

There are numerous performance metrics in the literature to assess different aspects of task performance. Determining which criteria to select and how to use them to evaluate a team's overall performance has been much debated in the literature (described in more detail in section 2.2). The general approach has been to assess the performance ability of each agent, characterized by the important performance parameters (completion time, reliability of completion, mean time between failures, resource utilization and cost, mental workload, etc.).

There has also been great diversity in the computational methods used to aggregate the results of the different individual task performance metrics. After the results for each task are tabulated, the next step is to rate the performance of the team as a whole in completing the overall scenario's goals. Several model types will be discussed in section 2.2.5 that demonstrate how to incorporate multiple performance metrics into an overall score.

The purpose of the team analysis is to determine the quality and efficiency of the team's overall performance, with the ultimate goal of enabling quantitative comparison between teams.

1.2.3 Incorporating Optimization to Improve Team Performance

Creating a mission plan for any space operations quickly turns into an over-constrained multi-objective optimization problem. There are many different kinds of constraints on the crew activity scheduling problem [52], including differences in crew performance capabilities (dependent on the crew member's physical support to a structure during the tasks), relational, topographical, and precedent constraints between tasks, and time window constraints for nominal EVA operations.

Team members can be used in various combinations to better utilize their capabilities and skills to create more efficient and diversified operational teams. This involves allocating tasks to provide the most benefit from the partnership, and creating the planning, scheduling, and software interfaces to support these efforts.

To simplify the overall problem into one that has a tractable solution, mission designers to-date have arbitrarily assigned importance to mission parameters, subjectively limiting the search space. While this has been effective at evaluating individual mission plans, the arbitrary evaluation criteria makes a straightforward comparison between two plans based on different ranking scales impossible. To facilitate comparison between different analyses, being able to specify an objective set of criteria would be immensely valuable. How could this be obtained? The intractable problem must be simplified to enable feasible solutions. The question then becomes how to select an objective set of criteria for any given problem.

Fong [29] described a proof-of-concept experiment to show that robotic reconnaissance missions could supplement human exploration. The concept was to send a mobile robot to scout out the terrain to aid in selecting a traversal path for a two human EVA team on the moon. The scout would collect environmental data that could be used to improve each day's mission plan and improve human crew performance by reducing the operational risk and increasing its situational awareness. Individual task metrics for the scout were reported in real-time to the operator, were compiled into overall metrics and the data was used by operators to adjust operations.

This type of overall mission analysis to differentiate team configurations could be an inordinately valuable tool for mission designers. The challenge becomes creating a quantitative, overall model to measure a team's performance for a generic mission, to enable broad use of the analysis tool (and remove the need for mission-specific models).

1.3 Unique Dissertation Contributions

A generic, objective methodology to select and aggregate performance criteria to aid in determining an optimal HRT configuration for a mission scenario will transcend these three domains. The HRT configuration selection problem is utilized as the application that motivates the problem. The methodology, however, will provide a much broader search of the design space between performance metrics and optimization models that facilitate evaluation of the design options.

The unique contributions of this dissertation are briefly itemized here, and discussed in more detail in Chapter 3:

- Decomposition of the HRT configuration selection problem into three distinct realms, decoupling their analysis: classification of agent and mission details, planning and scheduling, and selection of metrics and teams.

- Select performance metrics to preserve underlying dominance structure of a problem - this is a new approach for how to choose performance metrics.

- Demonstration of a method to compute overall performance evaluation that does not rely on aggregating multiple performance metrics into a single performance function.

- Create a generic, rigorous, objective, quantitative methodology for the HRT domain to both select the performance metrics to be used in an overall team performance analysis and to reduce the complexity of a mission designer's final decision space. This type of methodology is novel for the domain and a valuable contribution to further the use of human and robot teams.

Chapter 2

HRT Performance Analysis Literature Review

The literature review section of this dissertation contains an extensive survey of the field of collaborative HRT performance metrics, structured to answer the challenges laid out in the three domains of the problem statement (section 1.2). It assesses the different priorities, assumptions, and methodologies incorporated into the leading quantitative models, with specific emphasis on determining which are most applicable in spaceflight applications. It discusses the challenges of implementing the methods, the shortcomings of the methods, and possible adaptations to increase the generality of the methods.

This is a necessary first step to enable analysis of the effect of the different evaluation metrics on the design of team participation. In a unique contribution to the field, this survey (originally published at the 2011 International Conference on Environmental Systems [91]) sought to identify the conceptual pieces of existing HRI methods that would best synthesize into a universal, objective quantitative team task performance evaluation model, valid for a wide range of applications. It is this final purpose - creating an objective performance evaluation method and model - that is the aim of this dissertation.

This portion of the literature review focuses on monitoring and measuring the overall team performance of a human and robot system. For more information on research about measuring and improving human workload in human and robot teams, the author suggests reference [72] which contains a review of human workload models based on cognitive resource utilization.

2.1 Characterizing HRT Configurations

Three fundamental aspects that categorize a HRT are the ratio of humans to robots, the required amount of human effort to command and monitor the robots, and the capabilities of the robotic agents utilized. These greatly influence the types of coordinated tasks that can be done, and creates bounds on the communication architecture needed to provide efficiency in the team's cooperative operations. The realm of possibilities and combinations is virtually endless. In each of the following subsections (sections 2.1.1 through 2.1.6), the component pieces of a HRT architecture are considered, and the performance metrics that characterize the relevant parameters are included for coherency. Properties of performance metrics and how to use them in an overall team performance analysis will be discussed in section 2.2.

2.1.1 When to Analyze

Task performance analysis can be a useful tool during several different periods of mission operations. Experimental testing before and during the mission design phase provides mission planners with data on how well each agent could perform a wide variety of tasks in a future mission. The data evaluates each task separately rather than as part of a full mission. The data can be used during the design phase of a mission to determine which agent should perform each task (task allocation), and can facilitate arranging each agent's schedule to ensure that all tasks are completed during the mission and that the workload is distributed between all of the agents.

A priori analysis can also be performed to assess a team's ability to resolve anomalies during task performance. It is at this point in the design process that many different options can be considered, and comparisons can be drawn between different combinations of agents to determine the best performing configuration. This in turn will objectively select the final team members (agent selection), task allocation, mission objectives and task ordering.

Additionally, task performance analysis can be evaluated in real-time during a mission to assess how well the tasks are being completed. Real-time analysis requires a significantly different hardware architecture to support the quantity of computing and processing of sensor information that must keep up with the real-time operations task data. The advantage of real-time analysis is that details of team performance (including task allocation and anomaly resolution) can be addressed and altered as needed to improve the team performance for the rest of the mission.

Whether the mission scenario is in real-time or decided *a priori* has a significant effect on the task allocation, planning, and scheduling that occurs. Done *a priori*, these can be cycled through software to optimize the scenario over the entire mission length. Computational efficiency may still be an issue, but it is removed from being mission critical. In real-time, a team's task allocation, planning, and scheduling can be reassessed and changed depending on immediate performance needs and capabilities. If a task is taking longer than expected, other subsequent tasks can be reassigned as necessary to balance out the schedule to maintain efficiency. With distributed levels of intelligence, there may not be an initial team plan to deviate from. Decisions could be made as each task-need arises. In operations with supervisor decisions made *a priori*, none of these performance qualities can be altered mid-task, and must wait through the end of the task or scenario.

When to Analyze: Performance Metrics

A human supervisor can make decisions in real time, but there are limits to human information retention, which depends on the quantity to be sifted through, and the complexity and heterogeneity of the tasks. These lead to a limit to the number of robots that an operator can control simultaneously ("fan out" (FO) [61]) without degradation of team efficiency or under-utilization of team resources. An overloaded supervisor will develop a backlog of actions and decisions. If a robot has to wait for an operator's instructions, an anomaly resolution plan, or confirmation of any kind, this reduces the efficiency of the robotic agent.

Rather than relying on task completion times to measure how well an agent performs a task (as done in HURON), Schreckenghost [83] developed the concept of a work efficiency index (WEI). The WEI metric represents the ratio of productive time to overhead time that occurs in an agents' task schedule. Schreckenghost notes that WEI is more valuable in *a priori* analysis because it utilizes total productive time and total overhead time for a mission. Using these metrics, however, is not practical for real-time task performance analysis. The values could vary greatly between each real-time increment and could give a false representation of the overall efficiency of an agent.

2.1.2 Scenario Perspective

Whether a scenario is designed from a human-centered or robot-centered perspective influences which variables and resources are the most mission-critical and can change the assumptions behind the scenario development. For example, a human-centered model might assume that human extra-vehicular activity (EVA) time is the most critical commodity because of environmental exposure risks to the human outside of the spacecraft. This type of scenario would be planned such that the human agent is only used when absolutely needed. Alternatively, in a robot-centered model, the robot could be designed to be fully capable and independent of the human. Any evaluation of task or team performance would be geared towards evaluating efficiency and effectiveness with respect to the robot.

Scenario Perspective: Performance Metrics

One perspective to assess a team's efficiency, effectiveness, and productivity is by analyzing its resiliency to failure. Kannan [48] defined a metric for calculating how useful fault tolerance is for a multi-robot team. The paper provided a practical method to calculate the redundancy of a system. Shah [87] examined the productivity of a HRT through the mean time between interventions (MTBI), mean time completing an intervention (MTCI), and the probability that an intervention would be needed. It described the effect of unplanned interventions on a team's productivity, and demonstrated that the team's productivity was much more sensitive to MTBI than to MTCI.

Within each scenario perspective, the roles of the humans and robots can vary. The roles have significant effect on the way tasks are completed and the workload distributed. Scholtz [80] discussed five different roles that a human can have when working together with a robot on a task: supervisor, operator, mechanic, peer, and bystander. As a supervisor, the human gives direct instructions to the robot. The detail of the instructions (high level commands versus primitive level scripted tasks) depends on the robot's abilities to interpret and implement instructions. In addition, as a supervisor the human is responsible for forming and creating a plan to pursue the overall goals. A human could also be an operator who directly influences the robot's actions. As a mechanic, the human is collocated with the robot and participates physically with the robot's hardware and actions. Humans as peers to the robot are also collocated. In this role the human gives commands or information aid to the robot, but does not participate in robot upkeep. The final role of a human interacting with a robot is as bystander. In this role the human does not participate in any tasks with the robot. The human is only an obstacle in the robot's environment.

In Singer [88], the effect of redefining a robot's role on a HRT was analyzed. The roles were differentiated by the safety considerations that determined the ability of a robot to work within the human crew's proximity. In each role the robot exercised different physical capabilities that determined the portion of mission tasks that it could perform. With the overall objective of minimizing human EVA time, the role definitions were traced through the task allocation and scheduling process to determine how the different perspectives changed the overall team's performance and efficiency.

2.1.3 Hierarchical and Distributed Decision Making

Decision making for a HRT can either be hierarchical or distributed. Most HRTs to date have used hierarchical decision making, where a supervisor oversees the team and there can be several levels of authority. In distributed decision making, a team is made up of individuals who can each make task decisions for themselves, relying on their sensory information of the world around them, and relevant information passed to them by a neighboring agent.

Decision Making: Performance Metrics

In Fong [31, 32], the Peer-to-Peer Human-Robot Interaction Project was described. In the model, a task executive allocated tasks to agents which were both capable of performing the work and were not actively working on another task. A feature of this tool was that the task executive had the hierarchical decision authority to interrupt an agent's task and send it to perform another (preemption). Once a task was assigned, however, decision making was passed to the individual agent to plan and schedule its own task.

Ponda [71] described the development of a real-time decision making framework that allocated tasks for a heterogeneous HRT. The predictive model developed schedules for all of the agents based on agent availability, workload, and any coordination requirements between the humans and robots needed to complete the given task. As the number and diversity of agents in the combinatorial problem increased, centralized planning became computationally prohibitive. Using a decentralized approach (such as decentralized auction algorithms) to facilitate the task allocation reduced this challenge, and facilitated the generation of a more efficient and effective architecture.

Billman [8] presented an extensive table of performance metrics, their relevant parameters, and the human factors concerns associated with each. These were used in several experiments to evaluate a mixed-initiative (both robot and human can initiate communication and tasks) human-autonomous unmanned vehicle teams for the Navy's Intelligent Autonomy program.

Saleh [76] expanded Crandall's model [17, 15] to include two factors that represent the level of trust for both the human and the robot during cooperative interaction intervals. Trust in a HRT is representative of the human's belief that a robot understood its instructions, that it will correctly assess its situation, and that it will perform tasks correctly without requiring assistance. In Saleh's [76] work, a human's trust level was proportional to the human's fan-out (FO), and indirectly proportional to the robot attention demand (RAD). RAD was defined to be a function of both direct interaction time and indirect interaction time. Indirect interaction time was explained to be the interval when the robot is working autonomously but the human, due to reduced trust, monitors the robot's progress rather than applying the human's full attention to another task. Another factor gaged the level of human trust in the autonomous system making correct choices and following through with those choices. Decreased trust increased the time duration of the interaction.

Freedy [33] presented a methodology to assess trust levels for a HRT during operations. Expected value statistics were used to decide whether to allocate control to a robot. This work used a tool developed by Visser [21], the Mixed Initiative Team Performance Assessment System (MITPAS), which calculated a compound "goodness" score as a relative expected loss score, derived from observed human task allocation decision behavior, risk and observed robot performance. This was used to maximize the performance of the overall team. The MITPAS system was developed and validated for operator training on unmanned vehicles.

Marble [57] described the level of trust that the human had both for a robot making a correct decision, and in the robot completing a decided path effectively. This work assessed trust limitations in mixed-initiative systems. The operator's situational awareness and trust within an experiment in which operators directed mobile robots to perform tasks was measured for a given scenario under five different autonomy modes.

VanWissen [97] conducted a study to investigate how trust and fairness factored into human interaction with other humans, and with robotic agents. His results demonstrate how trust affects human decisions, preferences, and ideas of reward in a collaborative human and robot setting.

2.1.4 Autonomy Level

Robots have been developed with increasing levels of autonomy, such that they are able to not only carry out scripted tasks by themselves, but identify anomalies, and come up with their own resolution plans. Technology has developed not only to create distinct autonomy levels for a given robotic system, but also to allow adjustable, or sliding autonomy (allows the autonomy level to change as needed within a scenario). Miller [58] has done research with varying levels of robot autonomy to determine their effect on human counterparts. Heger [40] demonstrated that the concept of sliding autonomy could reduce the probability of an irrecoverable failure and would increase overall team efficiency.

Due to safety and reliability concerns, the majority of robots used to date in space mission applications (as peers to human agents) have been controlled by a supervisor (e.g. the Space Shuttle's remote manipulator system). For this reason, much of the literature presumes supervisory control of robots in a scenario that actively involves a human presence. In the scenarios where the human supervisor is off-scene, robots are designed with a higher level of autonomy (e.g. Mars rovers), allowing them to better analyze their own situation and pursue goals and way-points independently.

In Goodrich [37], design principles were developed that would guide the development of human-robot autonomy architectures to make interactions more efficient. More recently in Goodrich [38], the success of two operator management styles were described for a team of robots with adjustable autonomy. Usually supervised robots would each be given orders sequentially - the operator's attention would switch to a new robot only when finished with the current one. Goodrich proposed an alternative style which would direct the robots in a method similar to a sports playbook. After a play was announced, each robot would determine the course of its own actions to facilitate achievement of the team goal.

Howard [43] examined four different human-team leadership styles to identify their defining characteristics and adapt them to HRTs to create more effective configurations. Her analysis suggested more effective teams would be formed if humans used a directive style or a transactional style of leadership with their robotic peers. In a directive style, the human is the supervisor, giving commands and establishing the overall goals. In a transactional style, the human plays the role of the operator, monitoring the robot and its task completion, and giving commands at a lower level.

Metrics for Autonomy

Glas [35] discussed two new metrics to represent task difficulty for a human monitoring and controlling a multi-robot team. Situation coverage (SC) is the percentage (of total tasks) in which a robot understands the directive. In multi-robot situations, SC measures the upper bound on the ability of the system to operate autonomously. Critical time ratio (CTR) is a ratio of the time that a robot is performing mission-critical tasks to the total amount of time that a robot is actively engaged in tasks. When two robots have high CTR, an operator will have a higher workload. A human can improve performance by reducing the number of conflicts that occur between robot attention demands.

A new way of thinking about HRTs was developed by Olsen and Crandall to research the most effective ways for humans and robots to work together on tasks, specifically addressing autonomy levels. They developed a series of metrics that addressed the specific unique human and robot interactions at a generic level. In Olsen [61, 62, 63] and Goodrich [37], they introduce several new metrics. Olsen and Goodrich concretized neglect time (NT) as a metric to measure robot autonomy [61] to reduce interaction effort (IE) for both a human operator and the robot without reducing the effectiveness of the team.

This metric, however, considers human attention and does not consider the physical abilities, limitations, or usage of the human. NT is the amount of time a robot can function independently of the human. This interval is demarcated by a user-defined drop in the effectiveness of the robot's task performance to below a threshold, which can only then be raised by human intervention. NT and a quantity representing a human's interaction effort (IE) combine to calculate a quantity for robot attention demand (RAD), which is the fraction of the total task time that the robot requires attention. Fan out (FO, defined as the inverse of RAD), represents the number of robots a human can monitor and control before the decrease in overall team performance drops past a threshold. Crandall [18] used these metrics to predict the performance of a multi-robot team and validate the method with experiments taxing the operator's attention. In addition, this method found the performance thresholds that maximize team performance, given a team size [17].

Crandall [15] solidified these concepts into a methodology that has significantly affected the field. Wang [99] extended Crandall's NT metric to include the coordination demands (CD) and the effects of robot heterogeneity. Occupied time (OT) differentiates between wait times resulting from an operator having low situational awareness and wait times resulting from a queue of needy robots. Wang [98] expanded the estimation of an operator's FO limit to allow analysis for N-robot teams.

Elara [25] extended Crandall's model to include the possibility that a robot did not correctly interpret or respond to a user's command. In a false positive, a robot rejects a "correct" interaction. In a false negative, a robot fails to reject an "incorrect" interaction. False alarm time (FAT) represents the time that is spent identifying a false alarm and recovering from the delay.

In the two papers by Schreckenghost [83, 84] the authors described metrics and a quantitative model to assess real-time adjustable autonomy. They suggested measuring the degree of robot independence (to decrease human intervention time) based on the time spent on unplanned interventions. The computed performance metric results were used in real time by controllers. It should be noted, however, that results were based on percentage of mission time but that there was no indication given to relate robot to human time. If a robot performed its share of the tasks at a slower rate than in the previous scenario run, then all else being the same, it would register as spending more time in an autonomous mode. This could skew the results because performing tasks slower does not require a different level of autonomy, and should not be perceived as such.

2.1.5 Situational Awareness

Situational awareness, as discussed in the literature, can refer to three different perspectives: that of a human supervisor referring to knowledge of overall mission operations, that of a robot control operator referring to knowledge of the robot's immediate environment and obstacles, and of the robot's awareness of its own environment. Quantifying situational awareness has been a challenge because the idea has many definitions and contributing factors.

Situational Awareness: Performance Metrics

Bruemmer [13] proposed using human workload, error, and overall performance as quantities that gage the effectiveness of robots in a mixed-initiative environment. The amount of human error could be seen as a reflection of operator fatigue, but more dominantly a lack of human situational awareness of the task environment. The model created a text-based dialogue between the human and robot, and produced a 3-D representation for shared understanding about the task and the environment. While this research was intended to be a proof-of-concept argument that increased autonomy and better operator interfaces could improve robot navigation, it also demonstrated collaborative control where robots were effectively used as trusted peers.

Scholtz [81] described the implementation of a tool that evaluates the situational awareness provided to an operator through a user interface used for supervisory control of autonomous vehicles. Each interface was evaluated to find how well each facilitates a user's situational awareness.

Lampe [53] described a metric representing environmental complexity and a robot's information of it as a measure of robot autonomy. Nehmzow [60] presented a novel quantitative model of a robot's interaction with its environment based on the robot's trajectory over time.

Nehme [59] created a model to evaluate the performance of supervisory control of multi-unmanned vehicles. Included in the model were both operator variables and variables that pertain to the entire team. A unique contribution of this model was that it modeled operator limits by accounting for wait time due to loss of situational awareness. The integrated model facilitated comparing design development and was capable of searching the large design space to obtain the overall goal of maximizing team efficiency.

Hwang [45] developed a model that used each agent's knowledge of the other agent's state and of the environment's state to measure the changing interactions between the agents. This model can be used to measure the situational awareness of each agent, and the interaction effort required at each state of the cooperation.

2.1.6 Communication Architecture

For a heterogeneous HRT, creating a communication architecture that facilitates passing relevant information when needed, and avoiding information overload (passing unneeded sensory data) is imperative for an effective team. There are many different architectures that can be utilized in the communication framework, depending on the agents involved in a mission. In most supervisory control scenarios, human operators or peers communicate to the robots, but the robots are not capable of passing confirmation of task completion messages to the humans, and are not capable of querying their supervisors with questions or to address task difficulties.

Kaupp [50] presented a robot-centric model to facilitate bidirectional communication between a human peer and a mobile robot. Within this paradigm, robots passed images of the environment to the human to increase the human's situational awareness and to process details from the images. Humans were viewed as a resource which could be queried for information and observations about the environment, but as with all resource usage, it came at a cost. The cost of querying operators was traded off within the architecture against the expected benefit of the new information. This model could be used to determine what and when to communicate between a human and robot engaged in collaborative tasks.

Approaching the difficulties of team communication from a different direction, Trafton [95] presented a cognitive architecture to facilitate perspective-taking for collaborative human and robot interaction. The goal of the model was to produce intelligent robots that were capable of reasoning from a human-perspective by modeling how humans integrate multiple information and environmental representations into a world model. This type of reasoning allowed a robot to spatially interpret relative commands from a human, e.g. "give me the wrench on the right". To select the specified object correctly, the robot needed to be able to simulate the perspective of the human, which required implicit rotations of the world environment.

The Human Robot Interaction Operating System (HRI/OS) software [31, 32] was created with several distinct objectives. In addition to facilitating perspective-taking, the overall goal was to both maximize the amount of work done by the entire team and to reduce the number and duration of EVAs needed to complete the task list. To further reduce the human's workload in the scenarios, human and robot communication was designed to have as natural (for humans) interaction mechanisms as possible. This includes having a text-to-speech agent that verbalized a robot's responses for a human to hear, and a speech recognition agent that translated a human's verbal response into text for a robot to parse.

2.2 Challenges to Comparing HRT Configurations

2.2.1 How to Define and Assess Task Performance?

An innumerable variety of performance metrics have been identified to help define the phenomena that occur between a human and robot working cooperatively on a task. Some of the metrics are application specific, and some are more general. These individual metrics canvas the range of activities done by each agent. Common metrics include task completion time, reliability of task completion, mean time between failures, resource utilization and cost, mental workload, and a transition or switching cost for transferring mental attention from one task to another.

Burke [14] described application-specific HRI metrics for urban search and rescue that could be applied to more generic scenarios (search, rescue (extrication), structural evaluation, medical assessment and treatment, information transfer, command and control, and logistics). The project used existing models and software systems (Robot-Assisted Search and Rescue Coding System (RASCAR-CS) and the FAA's Controller-to-Controller Communication and Coordination Taxonomy (C4T), which capture what is communicated and how it is communicated for a team) to examine the robot's effects on human task performance within the context of human work. In essence, it logs how the robot affects and aids human performance.

Keller [51] presented a straightforward example of a human driving a car while talking on the phone to break down a multi-part task list to develop metrics that characterize the activities. The human's performance in the example was evaluated by considering a task as several simultaneous actions (visual, auditory, cognitive, psychomotor, all measured in relative rating scales). Activity in all of these pieces could lead to excessive workload demands, leading to performance errors (note, without any other agents taking part). This paper measured the scaled values of the number of human resource components needed for each task to simulate quantitative predictions of the human's workload over the entire task list.

Most models that attempt to measure the performance of a team including humans have a common metric for measuring human mental workload. The model that is commonly used is the NASA-TLX [39], which provides workload values for various human mental tasks. This results in subjective data from a post-experiment questionnaire to gage test subjects' estimated workload during different parts of an experiment.

An innumerable quantity of software planning packages have been developed to plan a HRT's operations. In general, each research group working on a mission scheduling problem has used their own software packages, and there is no community consensus on which is preferred. While each package is unique, most can be applied to the same types of scenarios. For example, an overall lunar mission planning software package (HURON (Human-Robot Task Network Optimization) [26]) has been developed at JPL to facilitate task allocation, planning, and scheduling for combined human and robotic activities. It provides the architecture to develop optimal task allocation and scheduling for a scripted multi-agent lunar mission. The input problem description in this software package is decoupled from the planning and agent assignment. This indicates that this software, like many of the other packages developed, can be applied more broadly than it was originally designed.

Arnold's work [4] created an analytical framework to compare the advantages and disadvantages of different human-robot systems. Given a set of metrics, the goal of the framework was to lead towards optimal task performance. Although tasks and schedules were primarily scripted, the framework provided a guide for how to incorporate unplanned interventions into the schedule.

Each metric that quantifies a portion of a HRT interaction provides useful information to a mission designer or supervisor. Although there are numerous metrics to describe components of a HRT architecture and assess an agent's performance on individual tasks, it is clear that one single metric will not sufficiently explain the team's task performance. Quite the contrary. Each of the metrics mentioned in this discussion cover a different aspect of a HRT's performance, but successful performance in one category does not necessarily entail good performance in the other categories. Several metrics would need to be selected to comprehensively determine a HRT's performance by analyzing the different facets of the team's configuration. The problem that remains, however, is how to integrate the results from multiple metrics into a meaningful overall team performance score. It is only with this type of rating system that two different HRTs could be compared to determine their relative performance on an overall mission, rather than on single tasks or other criteria.

2.2.2 Team Performance Metrics in Related Fields

There are many applications that use humans and robotic technology cooperatively to complete a task. Bechar [6, 7] and Oren [64] developed a methodology for a performance analysis of human-robot collaboration in visual target recognition tasks. It was based on a quantitative model [6] with four levels of human-robot collaboration, ranging from manual to fully autonomous. The metrics used were from signal detection theory: hit, false alarm, miss and correct rejection. These quantify the influence of the robot, human, environment, and task to determine the optimal operational level based on input parameters. Bechar has developed algorithms to facilitate real-time switching between different collaboration levels for the human-robot teams [94].

The development of new computer user interfaces (UI) encounter similar challenges to those of HRTs. Stanton [92] demonstrated three experiments to record the usability of the UI and how it affected the user's performance in the application of human supervisors directing urban search and rescue mobile robots. This paper collected the data and left selection of an evaluation method for future research.

Yanco [104] demonstrated a proof-of-concept method for evaluating user control interfaces from rovers at an artificial intelligence search and rescue competition. Yanco categorized the evaluation of UIs into six categories of methods: effectiveness, efficiency, user satisfaction, inspection methods, empirical methods, and formal methods. Analysis of the competition runs and scoring results in several concrete UI suggestions for the type of information that, if present to the user, would have improved task performance. Bruemmer [13] measured the usability of an interface based on how the human workload and human error affected the performance of the autonomous robot.

Glas [35] developed a UI to increase the performance of multiple social robots by designing the UI to provide information to the operator in an intuitive way, increasing the human's ability to monitor several robots while specifically controlling a different one. Task difficulty, as used in this paper, referred to a task being more difficult if a robot required a user's attention to help with the task. The UI can be used to reduce the amount of attention each robot needs caused by task difficulty (to be differentiated from attention needed to relay a directive).

Mobile robots have been used in several experiments to observe their interactions with autistic children [20]. As seen in the experiments, autistic children have found that interacting with a mobile robot is less confrontational and less threatening than interacting with a human. The rovers passively sat in the room with the children, until they were comfortable and curious enough to approach the rover. They played tag with the children, and carried on simple conversations. In this case, the robots and humans did not have a specific agenda besides exploring their interaction. Assessing the overall performance of the combined child and robot would require integration of several different metrics that gaged various aspects of their combined activities.

2.2.3 Quantitatively Comparing HRT Task Performance

The large variety of options available for selecting and organizing team members in HRTs raises the question of how to evaluate their performance in methods that transcend the individual details of a team: in essence, how to compare apples and oranges. One approach is to select an individual metric and apply it to different teams. This is not sufficient, however, because the different priorities of the teams cannot be reflected in a single metric. Additionally, if the robot of team A excels in a mission, but it is the human of team B that excels in the same mission, how do you objectively weigh the two options?

An integral step in producing the most efficient task allocation and team schedules depends on the selection of an objective set of task performance metrics to distinguish between each agent's performance, and to measure the effect of changing task allocation or task ordering on the completion of the overall mission objectives. Frequently, the metrics used to assess performance are built into the software planning and scheduling packages that automatically develop feasible mission plans.

To reduce the computational load, most software packages create schedules that observe all constraints, but often the software is written to only consider a single mission objective (e.g. reduce overall mission duration, minimize human time). A much smaller set of task performance metrics is required to analyze each agent's contribution to meeting the mission objective (e.g. minimize task completion time). While this analysis will produce good results, the other performance parameters that are neglected (workload, resource depletion) in the analysis might have significant affects that are not brought to light.

Categorization of HRT Performance Metrics

There have been several attempts recently to create categorizations for individual performance metrics. Categorizing the metrics would aid the search for commonality between them. It could then be anticipated that a selection of metrics could be drawn that span each of the categories and facilitates a wider understanding of a team's overall performance. One approach has been to create taxonomies for human-robot interaction to specify categories for relevant details. These taxonomies have emphasized creating a rubric for comparing and contrasting the design decisions of different HRTs.

Gerkey [34] presented a taxonomy of task allocation for multi-robot systems based on the relative utility of one robot performing a task. This paper provided formal categorization of the different application problems that can occur (single-task robots versus multi-task robots, single-robot tasks versus multi-robot tasks, and instantaneous assignment of task allocation versus time-extended assignment, which assumes some predictive knowledge of the tasks that will need to be completed in the future). Several different algorithms were proposed to efficiently calculate the task allocation for each of these types of problems, and the algorithms were compared for computation requirements, communication requirements, and solution quality. While this paper focused on purely robotic teams, its methodology can easily be extended from the heterogeneous robot teams considered to a HRT.

Yanco [103] created a taxonomy that emphasized the physical characteristics of a team's dynamics. Team composition (defined as the ratio of humans to robots and including details about the different types of robots) was a defining characteristic - a team of one human and one robot will vary significantly in its overall task performance from a team of one human and a two-robot team. The degree to which a robotic agent is dependent on a human decision maker (autonomy level) will also significantly influence performance. Does each robotic agent require human intervention at some point? Can it resolve problems by itself? Can a human operate more than one robot in a given scenario? Does a robot need to resolve conflicting instructions received from different human agents? How much required interaction is necessary between team agents to complete tasks?

Yanco also described the information type, variety, and level provided to operators to facilitate situational awareness in decision making (including sensor information available, sensor fusion, and data pre-processing). Yanco cites Ellis [27] for the time and space part of the taxonomy, differentiating into four categories: humans and robots functioning at the same time (synchronous) or at different times (asynchronous), and physically located in the same place (collocated) or at a separated distance (non-collocated). This can clearly be seen to differentiate between a robot teleoperated from a control room and a robot peer working along-side its human contemporary.

Yanco's taxonomy applies to a broad variety of applications. It also includes a subjective category to indicate the relative importance of a task's performance, termed its criticality. This allows differentiation between mission critical performance applications (urban search and rescue) distinguished from applications with less severe consequences for failing a task (robot soccer team).

Steinfeld [93] provided a generalized categorization for common metrics that apply to human-robot interaction. Metrics were split into three categories: human, robot, and overall system. Each of these categories contains metrics in five additional categories (navigation, perception, management (including task allocation, resource allocation, and coordination), manipulation (interaction with the environment), and social).

These three taxonomies highlight different aspects of a HRT. Gerkey's taxonomy facilitates task assignment depending on team configuration, but it does not include categories for the planning and scheduling portion of the design problem, including workload, environmental knowledge, completion times, etc.. Yanco's taxonomy emphasizes the interaction mechanisms between the humans and robots much more highly, but provides little guidance for task allocation metrics or the effect of a HRT with distributed decision making. In Steinfeld's taxonomy, the details of each agent's task performance are clarified, and the interaction mechanisms between them are accounted for, but does not offer suggestions on how to combine a selection of metrics (including overall team performance metrics and agent performance metrics) into an overall picture. This includes incorporating multiple overall team performance metrics into a single ranking. Each of these taxonomies has been successfully used independently, and can apply to a broad range of heterogeneous teams composed of humans and robots. A future analysis might find value in integrating them for a more comprehensive analysis of what occurs during HRT interactions.

A unique approach to categorize team metrics has been developed recently by researchers at the Massachusetts Institute of Technology. Building on their previous work [15, 24, 19, 70, 69], researchers developed the concept of metric classes under the application of human supervisory control. They addressed the problem of which metrics should be selected to completely assess team performance. The more metrics were included in an analysis, the more computationally complex the analysis became and, as pointed out in Donmez [24], using metrics that correlate to the same data can result in finding false significant effects in the data.

Categories of different types of metrics were created in Donmez [24] with the goal of facilitating selection of metrics to be applied to a HRT. The resulting guiding principle is that to efficiently and comprehensively measure the overall team task performance, at least one metric from each category should be selected. Crandall [16] applied the formal metric classes framework to the application of UAVs to create a methodology for developing a predictive model of the interaction between humans and unmanned vehicles.

Although there are several different methods for categorizing performance metrics, the same goal is sought by each: a logical progression between groups of metrics would allow a mission designer to select the most relevant from a group and facilitate including a wider range of metrics to reflect the overall scenario more accurately.

2.2.4 Building Performance Metrics into Quantitative Models

Individual performance metrics for components of a HRT provide insight into the different qualities of team configurations. What is lacking, however, is a method or framework to incorporate these metrics into quantitative models designed to determine the quality of and efficiency of a team's overall task performance. This type of system-level analysis could determine which team would perform better for a given mission scenario without needing to run additional simulations or experiments, and will result in a better, more comprehensive picture of what is actually occurring. Each model will have built into it assumptions, priorities, and differing methodologies. Differentiating which model would be preferred in different scenarios can be a difficult problem. The following discussion seeks to categorize existing models and describe their characteristics.

Without a formal framework, a designer seeking to compare different team configurations would need to compare numeric results from a multitude of performance metrics to discover which configuration is best across the board. Schreckenghost [83, 84] described a real-time sliding autonomy robot assessment tool that provided several streams of individual performance metric data to an operator, covering the important components of robot operations. This type of analysis of several metric data streams becomes much more complicated with the increase in metric set size.

The calculation of composite task scores are used in other models to break down the complicated analysis such that comparison between team configurations can be done by assessing a single value for each. Rodriguez [74] provided a justification for using a composite task score by comparing the analysis to the scoring of an athletic competition. A composite numerical score for the overall competition was obtained from the scores on each of the individual events. If different sets of events were selected, the results were different. If all events were weighted equally, the individual scores were summed. If some events were deemed more valuable than others, a weighting factor was used to represent the importance. The final result from the competition was then a single score for each participant.

2.2.5 Parasuraman's Model Categorization

There are many implicit choices made in deciding which model to use, including whether the level of automation is assessed (including the coordination demands placed on other agents), and which components of the mission schedule are most important to the analyst. In a frequently cited paper, Parasuraman [66] presented two evaluation criteria to be used for this decision. The primary criterion was the effect of a given design selection on human performance, e.g. how was the human's performance influenced by this selection? The secondary evaluative criterion included several additional important pieces of a model, including reliability and cost. These criteria were for a human-centered model of HRTs, and were intended as a framework to guide an objective selection of a HRT for a given application. Parasuraman [66] identified four categories that represent existing quantitative HRT performance models: task load models, expected value statistics models, cognitive systems models, and state transition network models. Each of these types of models combine task performance data for each agent and results in an overall scenario-level assessment of team performance. It is with these types of quantitative models that it becomes possible to compare, in essence, apples and oranges - it facilitates comparison of different team configurations to select the best overall team for a mission.

Task Load Models

Task load models evaluate the effort required for each agent to complete a task. The goal of task load models is to examine the effect of the tasks themselves on system performance, operator demands, and how task performance responds to a range of autonomy levels. This can be in terms of time as a resource (task completion time), resources used (either an agent's or cumulative for the team, e.g. power), or overall, repetitive, and fatiguing workload levels for a human. Selecting this model, the designer's goal is to distribute the total workload across the agents such that the total task list could be completed in the minimum amount of time, constrained by the finite amount of resources and agents available. This type of model can be used to compare which agent should perform portions of a task list, and to compare which agent contributes most to the entire mission. It therefore can also be used as a task allocation schema, and a method to objectively select amongst a set of possible agents for the final team configuration.

A common method to combine metrics for a task load model involves a pairwise comparison of effort and execution time for each agent. This approach has been used by many researchers, particularly to determine the final task allocation amongst a HRT. Howard [41] validated using a composite task score (termed the sequence execution parameter) as a task allocation scheme instead of individual task performance metrics in an experiment to guide a robot to a target location. The overall composite task score was run through a genetic algorithm in which the weightings in the fitness function changed with user feedback. In another model, Howard [42, 43] used a composite task score for each agent to balance performance score and mental workload for teleoperated and autonomous robot control of an autonomous rendezvous and docking scenario. All of these works used the composite task score for the final task allocation decisions.

Task load models can be the most straight-forward of the model types to implement because the data required for standard-type missions, primarily execution time and workload quantities, will be readily available from previous missions or testing. No extra experimentation will be needed to obtain the required inputs to the model. Each agent of the HRT will have known capabilities, and the quantized effort required to complete tasks could be estimated based on similar tasks.

These types of models are preferred in space applications that are planned from a human-centered perspective. As an *a priori* analysis, task load models can provide verification of the task allocation schema used, and the mission objectives can be easily interpreted into the model to produce desired results. It can be much more challenging, however, to use task load models if a wealth of performance data for an agent is not available, e.g. for a new robotic system. Task completion time is one of the primary criterion in the model. New technology or tasks that agents have not attempted before will require either significant experimental testing before it can be used in the model, or a method to estimate performance used instead. While pausing in the analysis to perform experiments to obtain necessary input data is not the most expedient solution to the challenge, it is the most rigorous answer. Estimating models are often limited by their level of accuracy, which can feed a sizable amount of uncertainty into the task load model, and could propagate through in unknown ways.

Expected Value Models

The second type of model relies on the expected value to be gained from an agent performing a given task. This is a common type of quantitative model and can utilize a nearly limitless number of individual metrics. Each task performed by each agent is assessed to determine the benefit gained from that task-agent combination, and the cost of that combination. The tabulated difference between the benefit and the cost is the expected value. Evaluation is often represented in the form of statistics, where the probability of benefit and cost is used, with reference to probability of component or task failure. Parasuraman [65] contains a more detailed set of different analysis schemes for expected value models.

Rodriguez [74] was the first to use an expected value model to compute a composite task score as a relative value with respect to a reference. According to his model, a 'reference' could be an agent, or a team configuration. The model can be used to compare performance between agents to aid in comparing heterogeneous agents or between team configurations to evaluate overall team performance. In this model, tasks and agents were assessed for relative task completion time and relative resource cost. Dissimilar metrics could be used because the model resulted in a matrix of dimensionless parameters. Each agent's performance ratios and resource cost ratios were summed for the entire scenario considered. The performance ratios were summed for each agent such that an agent would have a single score that represented the difference between the relative benefit and the relative difference in cost of using the resources.

Rodriguez's model was structured such that each task primitive emphasized a different aspect of human performance (cognitive, motor, and sensory skills). This model calculated the value added by using a given team instead of the reference team. The selection of the reference does not affect the results - the analysis is relative, and the same relations would be achieved if a different reference was selected. In this paper, task difficulty included aptitude of a given system with respect to a reference, and a relative amount of power, mass, or other resource needed to implement the candidate system. The composite task score represented the ratio of value added by using a specified system or team instead of the reference.

Tunstel [96] applied Rodriguez's methodology for composite task scores to monitor the navigation performance of the Mars Exploration Rovers, Spirit and Opportunity. Mann [56] suggested using MacKenzie's modification [55] to Fitts' law in information theory to use a less simplified version of Shannon's law of communication theory in Rodriguez's performance ratio equation. This equation change allowed ratios in Rodriguez's model that were nearly equal to and less than one (desensitizing the equation such that the relative value of an agent did not mathematically result in zero), which could feasibly occur in practical applications.

Kaupp [49, 50] used a composite task score's value of information to determine an appropriate level of autonomy for navigating a maze. Using the HURON software [31, 32] to plan a lunar mission, Elfes [26] sought to maximize the value per cost of a mission by analyzing the required input effort to the expected output benefit of each task.

Expected value models are structured to facilitate comparison between different candidate systems. They require approximately the same input data as task load models, but the data is used as relative values rather than directly. This provides more robustness to uncertainty in the data itself. These models can also be run with only a rough estimation of how systems or agents perform in relation to each other (if task and resource data does not presently exist), removing the need to run more experiments before beginning analysis. The application of the models are more flexible than task load models: rather than necessitating a quantized overall score (in which case the range of scores would need to be analyzed to determine if a gap between two values is significant or negligible), the scores are immediately referenced for relative comparison. Expected value models are best applied when the designer seeks to select a configuration from a set of possibilities. On the other hand, if any other summary data is sought (comparison of the types of workload, distribution of the workload, or duration of larger workload quantities), further analysis capabilities would need to be built into the models, or a different model would need to be used.

Cognitive Systems

The goal of the third type of model is to evaluate the effect of tasks on human mental processes. The methods and systems of information processing fall into this domain. This model applies to HRTs because it is well adapted to cover not only human information processing about individual tasks, but also the coordination demands of working on a cooperative task (either with a human or a robot).

These models are best used for detailed analysis of the effect of different autonomy levels, situation awareness, and communication architectures on a human's mental workload. Due to their emphasis (if not exclusively) on cognitive processes only, these models do not assess physical performance. System level HRT overall performance evaluations generally do not benefit from this type of model, but these models can be invaluable in the development and verification that a mission scenario's required workload level is feasible.

State Transition Networks

This type of model attempts to frame a sequence of tasks into the agent states required to perform them (e.g. observer, active physical participant, standby, etc.). Agents only change from one state to another when a task assigned to them requires a different form of involvement. Built into the model is the assumption that minimizing the number of transitions will reduce the workload required to perform a task list in its given order and improve the efficiency of overall mission performance. This type of analysis is valuable for tracing what causes the changes to an agents' state throughout a scenario, and the direct effect between the actions. Heger [40] used a state transition matrix to map the probabilities of each agent's success or failure at a given task, with a composite task score accumulating for each task attempted. The transitions in this model referred to the operator yielding control to the robot, adjusting the autonomy of a HRT to have the greatest probability of success, and a timing metric was computed to obtain the expected duration of the task list. Yagoda [102] modeled human-robot interactions using Petri nets to map transitions between a UAV operator's states.

State transition models facilitate analysis of human attention and mental requirements for task performance. Rather than assessing from a workload perspective (as done in cognitive systems models), these models can easily incorporate physical requirements and coordination requirements into the performance analysis. Composite scores can be obtained to compare how often and what type of mental and physical mode transfer must occur during a mission, but the emphasis on the number of transitions does not necessarily correlate to better overall HRT performance. The scores instead reveal the influence of switching states on each different agent during task performance. A different type of model would be needed to represent a team's overall task performance.

Generating quantitative overall performance models for a HRT has been a significant challenge. Designers have offered frameworks to direct analysis to a set of common metrics to facilitate comparison between disparate team configurations.

The majority of these frameworks lead to developing a quantitative composite score model to evaluate a HRT's overall performance.

Shah [86] described the initial development of a different kind of framework to walk a designer through the process of selecting a HRT and the task allocation process. Rather than using a composite score, the evaluation of task performance relied on specific methods to link together common metrics for a space exploration task scenario. Each section of the paper provided a literature review of commonly used methods. The framework proposed unifying the process of selecting a team design and assessing a common set of task-based metrics to allow comparison of disparate team performance. This is a generalized framework that is an option to be used either instead of the quantitative models described in this paper, or in addition to.

2.2.6 Implementation Challenges

Of the four types of models described by Parasuraman [66, 65], deciding which best reflects the desired analysis perspective for a given application can be difficult. The different types of quantitative models are not mutually exclusive, however. In fact, it can be productive to combine features from several of them to fully characterize a HRT's interactions. It may be advantageous to analyze the expected value of a team configuration based on a cognitive model, or to structure a task-load model into a state transition matrix. Kaupp [49] presented a method to select the autonomy level prior to deployment of a HRT that produced the highest team effectiveness for task-oriented information exchange on a HRT. This analysis included actual robot performance data, and resource costs. A composite task score was developed that included execution time, pairwise comparison of effort, value added, and a weighting for the final composite. Selecting the type of model or combination of models will have a significant affect on the results, and can either highlight or obscure significant interactions in an application.

Once the type of model has been decided, the kind of input data and level of accuracy of the data required should be addressed. The level of accuracy of the input data could have a significant effect on the predictive performance of the model. Low-fidelity estimations feed uncertainty into the model which, depending on dependencies and correlations between parameters, could propagate through the model in unknown ways.

To determine the relevant input data to the model, it is necessary to select the set of metrics that will comprehensively reflect the overall performance of the HRT. A designer must select a large enough set of metrics to cover relevant aspects of the agent interaction, but must also avoid selecting too many metrics, which runs the risk of generating false correlations by analyzing the same effect from multiple angles. It is at this point in the model creation process that the designer must input details about the application, including the relevant subtask performance parameters, and the architecture details that facilitate the HRT's task completion.

The next challenge to implementing a quantitative model is explicitly framing the performance metrics in mathematical expressions that do not over-simplify the situation. For example, it would greatly simplify the modeling process to exclude environmental parameters from an analysis of how well one agent reliably follows the cooperative team plan. It could be assumed that an agent's comprehension of the task plan, and ability to direct its own efforts to achieve the desired goal would have a greater influence on the agent's successful performance than whether the terrain is grass, cement, or had vertical displacement. This simplified analysis would be fairly accurate in the majority of modeled cases, but it would diverge from the true behavior in cases where the environment has a significant effect on the agent performance. It is that very divergence that would be invaluable to have reflected in the model. It is necessary to ensure that the model includes all relevant dependent correlations and parameters to accurately reflect the true system performance.

2.3 Incorporating Optimization Into a Team Performance Model

After selecting the metrics to be included in the model, a designer must then link them to facilitate ease of comparison between results from different team configurations. Most of the researchers who utilized a composite task score as their overall team performance ranking [26, 42, 49, 50, 56, 74, 96] computed this value by a linear summation of the individual performance scores (or ratios, or expected values) for subtasks. It is possible to use weightings in the summation to include a measure of relative importance between the performance metrics. In cases where weightings were not specified, each metric had equivalent importance in the summation. These weightings will make a significant difference on the optimal designs returned from the system analysis. In essence, these calculations use the individual performance metrics as objective functions, transforming the evaluation of the HRT's performance into a constrained multi-objective optimization problem.

By combining the metrics with weightings into a single summation, however, the designer applies a common optimization technique to transform a multi-objective optimization problem into a single-objective optimization problem, which is much simpler to solve. To do this, knowledge of designer preferences between metrics or an estimate of relative weightings between the metrics is needed.

The weighted-sum method of aggregating objective functions is a specialization of a larger, more general field in optimization theory: utility function or value function methods. These methods combine multiple objective functions by developing relations between the objectives (or interactions between the objectives). These relations can be linear or nonlinear, but do require setting parameters to concretely define these relations.

There are several limitations of this method that make it less than ideal for application to the HRT configurtion selection problem domain of this dissertation. A utility (or value) function will return only one single answer (or one solution at a time). The function itself would need to be altered (the parameter values and relations) to find other Pareto-optimal solutions. In other words, the existence of multiple solutions on the Pareto front would not be detected by this method (multiple equally-performing team configurations would not be found by using this method - only one arbitrarily selected team would be chosen). Furthermore, the resulting single solution has no guarantee to be any better than other solutions not selected. The selection of a solution would be entirely dependent on the utility or value parameters used to define the relations.

Additionally, the utility (or value) function method requires users to develop relations between the objectives that apply for the entire design space. In the past, these have either been subjectively selected or all metrics assumed to be of equivalent value. There has been significant effort to make weightings selection more objective, but it has been diffcult to extrapolate these methods into the practical HRT configuration domain. Rohrmuller [75] presented a quantitative method to map the probabilistic interdependencies between individual performance metrics, and use them and their provided data to determine relationships between the metrics, and to compile them into a composite score that could be used to predict system performance and optimize performance parameters.

For further guidance on creating this objective method, the author suggests the extensive literature search by Bobko [9] which provided a cross discipline analysis that considered the validity of using weightings to aggregate subscores to represent a data group. Several methods were described and the reader was directed towards other references for more detail. If an objective method of selecting these weightings could be found, it would add rigor to the methodology of using composite scores to optimize a team configuration.

2.4 Conclusions from HRT Performance Analysis Literature Review

Future space exploration will involve humans working much more closely with robotic technologies, both as tools to ease the humans' workload, and as peers to expedite mission scenarios. Design options for creating the architecture of future HRTs have been explained, with sample performance metrics given for each category. With the numerous options available, frameworks have been described to guide selection of the most relevant metrics for each specific application. Incorporating several metrics into a quantitative method to facilitate comparison between different HRT configuration solutions will be invaluable for future mission design. To this end, the significant work in the research community to facilitate and develop quantitative models to calculate the overall performance of a HRT was analyzed.

The purpose of this dissertation is to continue to prepare the groundwork for the synthesis of existing methods to compare HRTs and measure the differences in their interactions. The presence of optimization theory beyond manipulating the multi-objective optimization problem into a single-objective optimization problem in the domain of human-robot interaction research is sparse. It is this area in particular that this proposed dissertation research will contribute significantly. A universal quantitative team performance model with a wide range of applications still eludes the research community, but work continues to progress towards this goal.

Chapter 3

Research Methodology and Algorithms

In my approach to the HRT configuration selection problem, I have identified three orthogonal axes of research. The first area of research has been in characterizing and increasing the diversity of human and robot teams. The state of this field was described in detail in section 2.1. Second, there has been significant research on methods to measure and quantify differences between each agents' performance and between overall team performance in various aspects of task completion. A substantial review of this body of research was provided in section 2.2. As described in section 2.3, there has been sparse application of advanced optimization theory in solving the HRT configuration selection problem - the third research axis.

This dissertation research sought to create a methodology that would increase the ability to quantitatively compare solutions in the design space of the HRT configuration selection problem. This was achieved by synthesizing optimization techniques to treat performance metrics as objective functions in a multi-objective optimization problem. The methodology will facilitate comparison between distinct groups of team members, and will facilitate selection a pseudo-optimal team for a given task scenario.

3.1 Background to This HRT Research Problem

For the last several years, I have been researching various aspects of HRT cooperation. The research has been along the HRT configuration and performance metrics orthogonal axes. My previous published research ([89], [88], [90], [91]) present a persuasive argument and supporting evidence that future space operations would benefit from involving cooperative robotic team agents in addition to the human crew. The papers were based on the same assumptions and the same framing of the HRT problem, allowing direct comparisons between the results in each paper. These papers define and assess how robots and humans can work together cooperatively to complete tasks, and address several critical issues about their combined performance.

In terms of addressing the challenges to comparing HRT configurations, the methodology developed in these papers provides a guideline for comparing HRTs before the mission design phase and facilitates quantitative comparison based on crew time as the primary criterion. This selection was guided by consideration of team efficiency from a task load model perspective. For a summarized discussion of the methodology assumptions, procedure, and results from this research, refer to the Appendix section A.1.

Three questions remain from previous research and have been the guiding structure of this dissertation. Is there an objective quantitative method to determine if a neglected parameter would have produced significantly different results? Is it possible to objectively reduce the problem design space to a core of important parameters? With an infinite number of ways to reduce the complexity of a problem, how could a designer objectively determine which is an optimal HRT configuration for any given mission? These are the questions that this dissertation research sought to answer.

3.1.1 HRT Configuration Problem Decomposition

In a unique approach to the HRT problem domain, I have conceptually partitioned designing a human-robot team into three distinct, sequential components, as seen in Figure 3.1. The first realm ("Classification" in the Figure) is where problem definition occurs. Mission goals and objectives are identified, and agent details, task details, and performance estimates and resource usage are collated. The second grouping ("Planning" in the Figure) encompasses all of the planning that occurs to build a mission from the requirements and constraints laid out in the first stage. Tasks are allocated between participating agents, schedules are developed, and resource limitations are imposed. This can be done intelligently using preferences, or this could be done by brute force to generate all possible options.

It is in the third stage ("Selection" in Figure 3.1) that this dissertation provides a unique contribution to the field. With the multitude of task allocation, team composition, and scheduling options, how could a mission designer objectively decide which is the optimal team for a given application? Fundamentally, this stage answers two questions: 1) How to objectively select an objective function set to use in evaluating team options? And 2) How to objectively select the best team for a

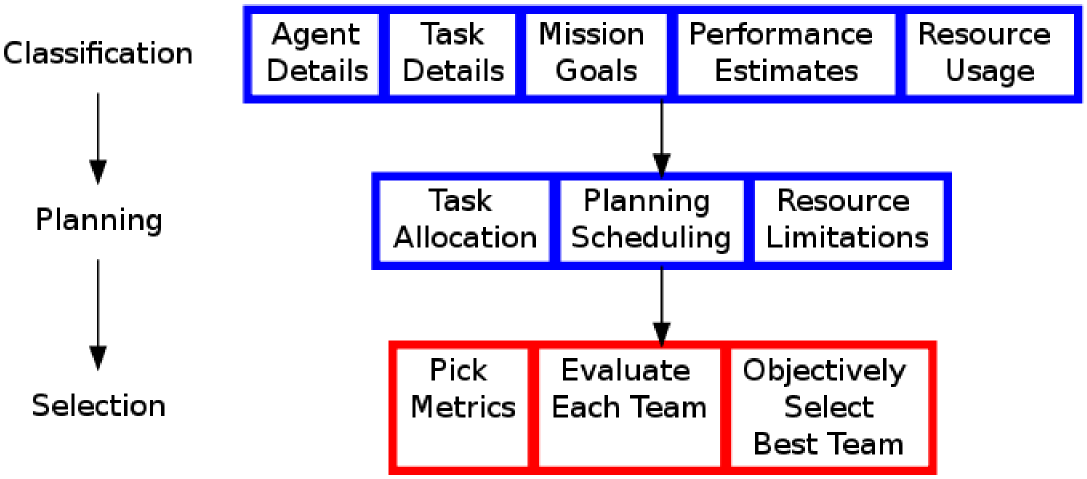


Figure 3.1: The Three Stages of the Collaborative HRT Problem

mission based on these objective functions?

3.1.2 Performance Metrics for the HRT Configuration Problem

The first step in the HRT configuration selection problem is to enumerate possible performance metrics that are relevant to the problem or mission scenario. It would be a Herculean effort to enumerate all possible objectives and constraints for the problem. It has been common practice in application to arbitrarily select those that the designers judge most relevant. As would be expected, the solutions from the problem will vary with the selected objective functions. Any decision to leave off a metric could potentially reshape the problem domain.

In the HRT configuration selection problem, there are many objectives that could be selected by a mission designer, and it is not necessarily straight forward to select some and neglect others. This research methodology will provide criteria to select a set of performance metrics to describe an overall team's performance and how this will affect the resulting team selection for a mission. This in turn will lead to a reduced set of performance metrics (objective functions) with which to compare different configurations of the same mission scenarios. A more detailed discussion of the rigor behind this set reduction can be found in section 3.3.

If a mission designer has a preference between agents or has knowledge of the relative importance of the performance metrics, the problem can be reduced to a single-objective optimization with known weightings. There are many established optimization methods to solve this type of problem (refer to section 2.3, so this variation was not assessed in this research).

3.1.3 Top-Level Research Outline

It is unrealistic to use all possible metrics in any analysis. To reduce this set to a tractable problem, this research sought a general, objective method to select a subset of the available performance metrics that have the most influence over the problem. It is this component of the process that is a unique contribution to the field.

Figure 3.2 provides a top-level overview of the different components of the methodology proposed in this dissertation research. Each of the components outlined in a bold-face color will be discussed in greater detail in sections 3.2 and 3.3. The green boxes represent algorithms from the literature that were implemented in Matlab for this dissertation research. The blue-outlined oval represents an existing software tool (a Matlab toolbox) that was used extensively in this research. It is the synthesis of these four components that represent the proposed methodology

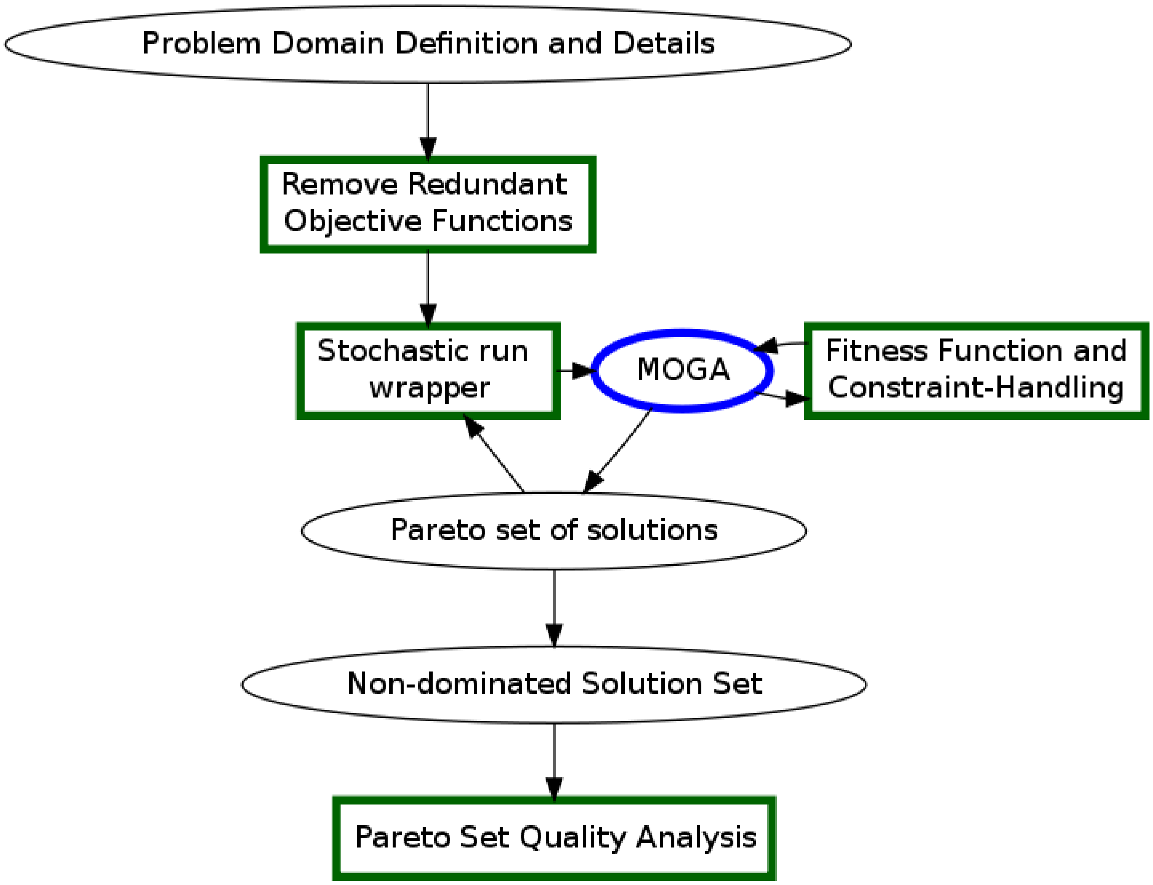

Problem Domain Definition and Details
Remove Redundant Objective Functions
Stochastic run wrapper
MOGA
Fitness Function and Constraint-Handling
Pareto set of solutions
Non-dominated Solution Set
Pareto Set Quality Analysis

Figure 3.2: Top Level Outline of Proposed Methodology

for how to objectively reduce the problem complexity of the HRT configuration selection problem and forms the unique contribution of this dissertation research. The synthesis of the components itself is not what makes this research dissertation-worthy - it is what is accomplished in the HRT domain that provides the valuable contribution.

There has been significant research in the area of reducing the dimensionality (or complexity) of multi-objective optimization problems (see section 3.3 for a more thorough discussion). Following the example of this field, an objective reduction algorithm was selected and implemented to enable selection of the most influential objective functions in a quantitative manner. The implementation was validated in the knapsack problem domain. The resulting solution set was compared to that obtained by a more traditional multi-objective optimization solution technique for several knapsack problem instances.

This methodology (using the objective reduction algorithm to reduce the objective functions of a multi-objective optimization problem, and an analysis of the resulting solutions) was then used in two different applications of current HRT research. In both cases, the goal was to analyze if this research's proposed methodology achieved different results than the more traditional solution methods. The first, a simple case study of reconnaissance rovers (modeling this research's analysis on the sample problem described in [96]), was designed to test the common performance analysis methods of other researchers and compare the results to those achieved using this methodology. After demonstrating its utility on smaller-scale HRT problems, the methodology was used on a demonstration large-scale HRT configuration selection problem. This final test case demonstrated how the methodology could help *a priori* decision makers evaluate and select a final configuration from a complex set of possible HRTs.

This steps beyond the question of how to pick the best team from the multiple options without more information. Is it possible, and how could a designer claim that the described optimization problem and its solution set represent an objectively better definition of the problem than others? Which are the best overall solutions to the problem – those resulting from objective set A or from objective set B?

Fundamentally, this is a question of how to create an objective methodology to select HRT performance metrics (objective functions to gage task performance). This involves evaluating how the selection of a set of objective functions and constraints affect the resulting solution set with the intent of providing a framework to simplify the problem without losing important problem information. It is this type of objective methodology that this dissertation research has created.

3.1.3.1 Representing Metrics and Problem Details

The first step in applying the developed methodology (as seen in the outline in Figure 3.2 and more detailed in Figure 3.3) is to enumerate the possible metrics. This requires representation of real-world concepts as mathematical expressions dependent on a common set of design parameters. The concepts Rodriguez [74] developed for using performance and resource ratios in a combined task load model and expected value model will be used in this analysis. Following Rodriguez's exam-

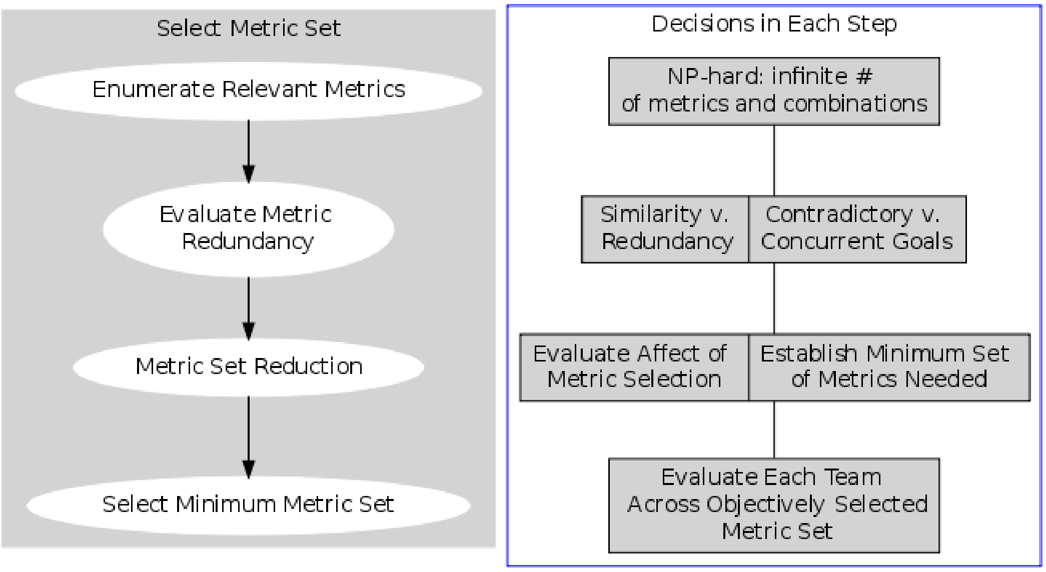

Select Metric Set
Enumerate Relevant Metrics
Evaluate Metric Redundancy
Metric Set Reduction
Select Minimum Metric Set
Decisions in Each Step
NP-hard: infinite # of metrics and combinations
Similarity v. Redundancy
Contradictory v. Concurrent Goals
Evaluate Affect of Metric Selection
Establish Minimum Set of Metrics Needed
Evaluate Each Team Across Objectively Selected Metric Set

Figure 3.3: Methodology Development for Performance Metric Selection

ple, mathematically representing different performance quantities can be simplified by using non-dimensional parameters rather than keeping track of and comparing different units. This also facilitates using different quantities within the same expression (rather than adding a time unit with a distance unit, a ratio of time to a reference and distance to a reference allows the unit-less values to be added).

Resources required for task completion should be represented both as an objective function (seeking to minimize resource usage to enable a larger range of tasks) and as a constraint (limited resources available). Other objectives such as mental workload of the EVA astronauts, will require analysis of input task performance data.

Performance metrics often measure similar variables or effects. The likelihood that some of them will overlap in their analysis is fairly high. The performance metrics should be evaluated to see if there is redundancy in the criteria used to estimate performance. As described by Donmez [24], using metrics that correlate to the same data can result in finding false significant effects in the data. Performance metrics should be analyzed to differentiate between similar (acceptable if they assess different qualities) and redundant objectives (potential trouble). Very different performance metrics can also have concurrent goals such that improving performance in one metric directly correlates to improved performance in another.

The goal of this analysis is to reduce the complexity of a problem by pruning the performance metric set to a minimal set. It is with the reduced metric set that each candidate team configuration can be objectively compared.

3.1.3.2 Additional Experiment Questions

There are several additional experiment questions that will be considered in the analysis. Is there a relation between the number or kind of metrics and either their convergence rate or convergence success? At what point (with how many metrics) does the problem fail to converge? How does including redundant or overlapping metrics change either the team selection or the magnitude of variation between the teams? A comparison of team combinations that achieved the same solution results would also be illuminating. All of these questions seek more information about the interplay of the design space and would lead to more information that could influence a mission designer's decisions.

3.1.4 Conceptual Simplification – The Knapsack Problem

To aid in an initial analysis of the proposed methodology, the human-robot team selection problem (an over-constrained multi-objective optimization problem as defined by Kurtzman [52]) can be compared to the knapsack problem. This simplifies the concepts involved without losing details in the analogy, and will be used in this research to simplify the real-world problem during methodology development and validation. The knapsack problem can be expressed as a set of problems with known solutions. This research methodology can therefore be verified in a domain with known solutions before being applied to a more complex problem domain.

Referring back to the three orthogonal axes that represent the HRT configuration selection problem (see Figure 3.1), the "classification" stage represented all of the problem details. In this conceptual simplification, the complexity of the HRT configuration selection problem is removed from consideration, and the known knapsack problem is dropped into its place.

The knapsack problem provides a conceptual simplification that maps easily to the large-scale HRT configuration selection problem. The knapsack problem has a list of candidate items that could be selected to be placed into the knapsack (a fixed quantity of and types of items to be placed into the knapsack). Similarly, the HRT configuration selection problem has a fixed quantity of candidate teams that can be selected. Imposing volume and weight constraints on the knapsack problem maps to implementing resource and usage constraints in the HRT configuration selection problem on each of the agents and available consummables.

For the knapsack problem, the goal is to identify which items to place in the knapsack to maximize overall profit. For the large-scale HRT configuration selection problem, the goal is to identify the candidate team configuration that maximizes overall team performance. Rather than being concerned with multiple different kinds of objectives (such as task allocation scheme differences between candidate teams), the knapsack problem's simplification considers only overall profit values for each item to be packed (or the performance capability of each team agent), which can be summed in a linear manner to aggregate into an overall perspective on total profit (or overall team performance).

A limitation of analyzing the HRT configuration selection problem from a knapsack problem approach, however, is the limitation in the variety of analysis that can be performed. The knapsack problem weighs each of the objectives equally and sums their results for an overall profit value. In the HRT configuration selection problem, it cannot be assumed that each of the performance metrics (objective functions) can be weighed equally. The nature of the performance metrics is not additive - they should each be optimized for their own merits, and an overall team analysis evaluated to determine how well each team configuration meets the requirements.

The knapsack problem will be used to illustrate the interdependencies of the objective functions (performance metrics in the HRT problem) and their effect throughout the design space search. The knapsack problem will be used as a reduced order application domain to verify that the methodology works and to calibrate the technique before returning to the HRT configuration selection problem.

The knapsack problem description has many similarities to the HRT configuration selection problem. Including more or less detail changes not only the problem description, but can have vastly different effects on the resulting solution set. For example, in the standard knapsack problem representation, when selecting items to take from the wealthy house, the thief assumed that it would take no effort on his part to sell all of the items that he can fit in the knapsack. His workload for collecting the profit was negligible. If, however, the problem description was expanded to include the effort the thief must expend to take the items to different locations to sell them, and any and all arrangements that must be made, it is likely that an entirely different set of items would end up in the knapsack.

This highlights the importance of fully describing all relevant portions of the problem. This dissertation's goal is to address this very problem – how to ensure that all of the relevant problem pieces are included in the objective functions and constraints while not overloading the problem with irrelevant data and detail.

A remaining issue to be discussed is how to know if the solution set resulting from a problem description is a good, representative set. Assuming that the returned Pareto set from a multi-objective genetic algorithm (MOGA) was accurate and complete would be faulty. It is for this reason that the methodology development will occur in the knapsack problem domain, as explained in Figure 3.4. Problems with established, documented solution techniques will be used to compare how the reduced objective function sets affect the solution sets.

It will be productive to compare the solution sets that the methodology generates for the knapsack problems to their established solutions. It is not necessarily the case that the methodology-generated solutions will be identical to the solution

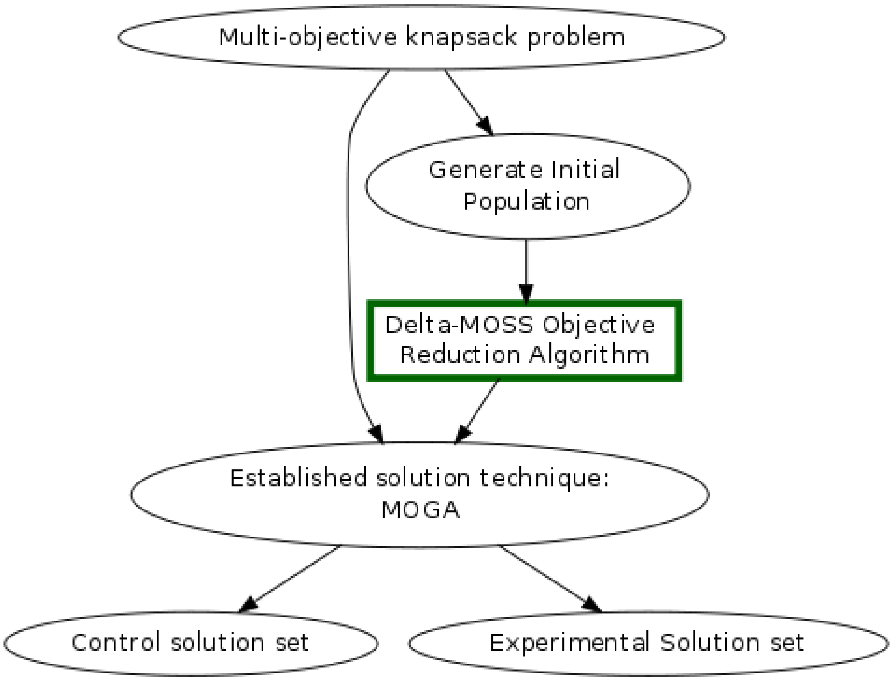

Multi-objective knapsack problem
Generate Initial Population
Delta-MOSS Objective Reduction Algorithm
Established solution technique: MOGA
Control solution set
Experimental Solution set

Figure 3.4: Methodology Validation in the Knapsack Problem Domain

sets generated by a more traditional solution technique. The methodology will be solving the problems from an entirely different perspective. If the methodology-generated solution sets can be verified by a more traditional solution technique, that would be ideal. If they are not, the methodology will still produce meaningful solutions.

3.2 MOGA Algorithmic Approach

To solve the constrained multi-objective optimization problem, Matlab's multi-objective genetic algorithm toolbox was used to come up with the optimal configurations for a given application and specific scenario. Matlab's multi-objective genetic algorithm toolbox was a tool used to generate new populations of data. A

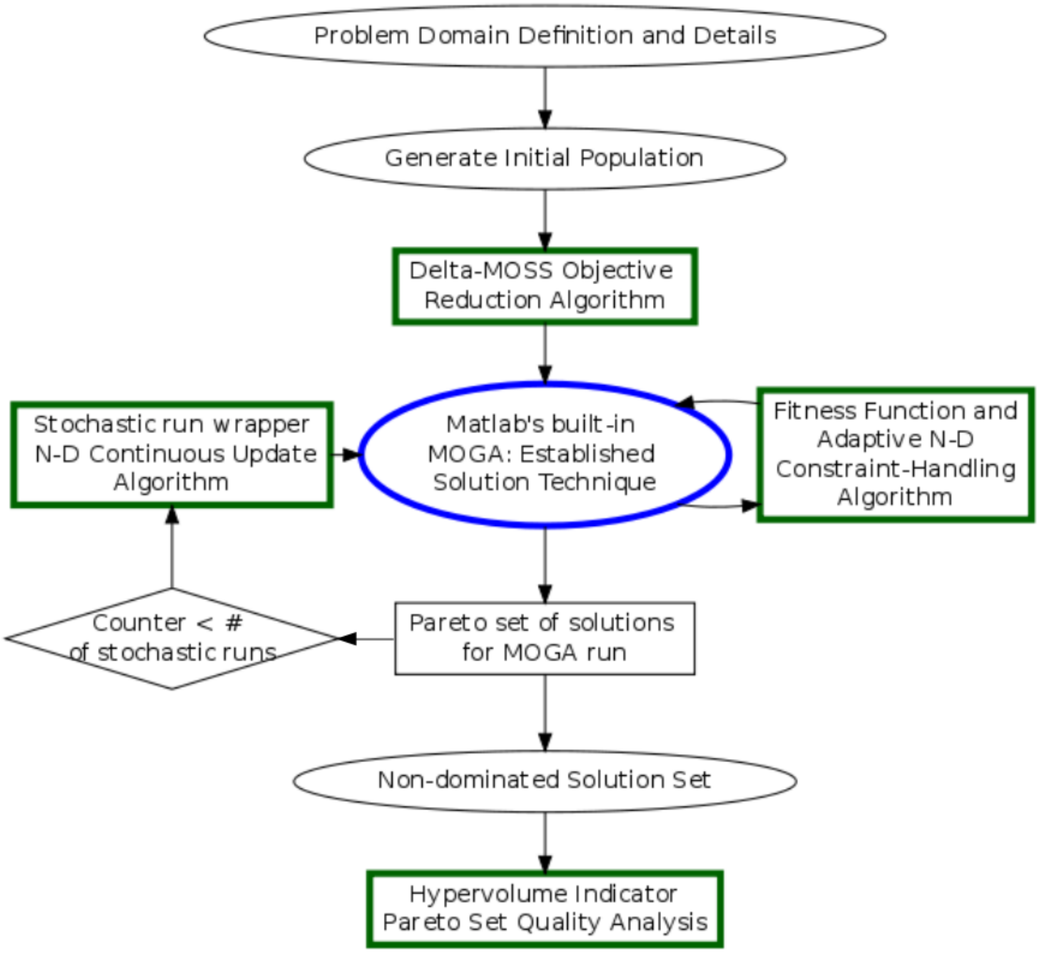

Problem Domain Definition and Details
Generate Initial Population
Delta-MOSS Objective Reduction Algorithm
Stochastic run wrapper N-D Continuous Update Algorithm
Matlab's built-in MOGA: Established Solution Technique
Fitness Function and Adaptive N-D Constraint-Handling Algorithm
Counter < # of stochastic runs
Pareto set of solutions for MOGA run
Non-dominated Solution Set
Hypervolume Indicator Pareto Set Quality Analysis

Figure 3.5: Outline of the Algorithmic Approach of This Research

user-defined fitness function contains all of the problem-specific parameters. Figure 3.5 outlines the algorithms implemented in this research to combine with Matlab's MOGA routine. Several existing algorithms from optimization literature have been implemented to incorporate multiple constraints into the multi-objective function evaluation and to account for potential search problems. Each component will be discussed in this section. Each algorithm is discussed below.

The results from this type of analysis yield a Pareto set of non-dominated solutions. This solution set represents different design options that are equally capable, according to the defined objectives, at completing the specific mission scenario. This solution set, however, only applies to the specific problem described by the mission requirements. If other objectives had been selected, it is highly possible that an entirely different solution set would be the result.

3.2.1 Use of Matlab's MOGA Toolbox

The role of Matlab's MOGA toolbox in this research effort should be put in perspective to clarify how it contributes as a tool to the overall methodology, but does not in itself represent anything more than number crunching. Figure 3.5 provides a visual representation to demonstrate where in the overall MOGA implementation the toolbox is used. Each run begins with the fitness function's evaluation of a generation of design points, which includes the objective functions, constraints, and the user-specified MOGA evolution parameters.

These fitness values are fed into Matlab's *gamultiobj.m* function to create a new generation based on the fitness of the previous generation's design points. This is iterated either for a specified number of generations, or until a tolerance between generation fitness is reached. In the former the algorithm does not necessarily converge, while in the latter the algorithm's convergence is the condition that terminates the MOGA. If the algorithm converges, the resulting generation is the Pareto set of solutions, where each solution is just as optimal as any other solution in the set, with none being better than any other.

It should be emphasized that Matlab's built-in toolbox is used solely to create a new generation of solutions for consideration. All other portions of this analysis process represent synthesized algorithms from other researchers.

3.2.2 Stochastic Analysis of a MOGA

A single run through a genetic algorithm might not search the entire design space. It is possible that the algorithm could become stuck in a local minima, resulting in non-reproducible results and sub-optimal solutions. To avoid this type of narrow-search complication, a stochastic wrapper function was written for this research to run the MOGA multiple times, keeping track of the non-dominated solutions across all of the runs. The wrapper function has the ability to call the multi-objective algorithm any number of user-specified times (10 is recommended as a general rule-of-thumb for statistical relevance). This stochastic solution set better describes the design space. Over the iterative runs, a larger swath of the design space will be searched, and the algorithm has a better chance of converging on a steady state solution set.

The stochastic wrapper assesses if any of the solutions from each run of the MOGA dominate each other. To achieve this goal, this research implemented the continuous update method for non-dominated solutions from Deb [22]. For use in this research, this algorithm was expanded to accommodate multiple objectives (n-dimensional) during the domination evaluation (the original continuous update method was specified only for the simple 2-dimensional case with two objectives).

The continuous update method is a faster computational method that does not check all solutions before deciding domination. Instead, it keeps a running list of non-dominated solutions. When an item in this list is dominated, it is removed from the list. When a new design point is demonstrated to be non-dominated by any other solution in the data set, then that point is added to the non-dominated list. After each stochastic run, the new Pareto set is compared with the existing non-dominated set, and the overall set of non-dominated points results.

3.2.3 Multi-Objective Constraint Handling Technique

Single-objective optimization problems have straight-forward options for incorporating a penalty function into the objective to handle the constraints simultaneous with solving the problem. A more nuanced technique is required for the case of multi-objective optimization. The main difference between all of the available techniques is how infeasible solutions are treated during the optimization process.

The simplest approach is a static penalty function that assigns a constant penalty to the fitness of all infeasible solutions. Frequently, this results in the infeasible solutions being disregarded in the evolutionary algorithm. It could be desirable, especially in situations where there are few if any feasible solutions, to use the information contained within infeasible solutions to further the algorithm. When too much emphasis is put on infeasible solutions, however, the time to convergence for an algorithm increases.

Woldesenbet's [101] adaptive multi-objective constraint handling technique has been implemented for this research, and is called from within the MOGA's fitness function. The primary advantage of this algorithm is that the relative importance of a given solution's fitness and the value of constraint violations is altered in each generation based on the number of feasible individuals in the given generation. In this way, the algorithm adapts to use infeasible solutions (by de-emphasizing constraint violation) when there are few feasible solutions. This adaptive penalty function does not require problem-specific parameter tuning.

3.2.4 Hypervolume Indicator Pareto Set Quality Metric

As seen from the convergence success and run time analysis from the previous sections, from a purely computational time perspective, the methodology proposed in this research does not seem beneficial. However, an analysis of the resulting Pareto solution sets will bring the utility of the method into greater light. It is an established practice in optimization theory to use quality metrics to gage the merits of a Pareto set according to two parameters: the convergence of the solutions to the Pareto front and the diversity of the solutions in the Pareto set. This type of metric enables comparison between two entirely different Pareto sets to gage the goodness for the solution set.

To evaluate the distribution of solutions along the Pareto front, n-dimensional quality metrics were used. A common quality metric in this regime is the hypervolume indicator (originally proposed by Zitzler and Thiele [107]). Bader [5] states that this is the only quality indicator that can be demonstrated to be completely sensitive (and correspond to) Pareto dominance, such that a higher hypervolume value of a first Pareto front with respect to a second Pareto front indicates that the first front dominates the second.

An additional advantage of using the hypervolume indicator as a Pareto set quality metric is that each front can be assessed independently of other sets as easily as its relative coverage can be compared (a valuable quantity both relative and absolute).

Zitzler [105] proposed using two complementary quality measures for the hypervolume calculation: the S and D functions. The S function is a measure of the volume of the objective space weakly dominated by a Pareto front (the size of the dominated space). The S value of two different Pareto fronts cannot be used to determine which set entirely dominates the other. A second measure is needed for the relative comparison.

The D function represents volume coverage difference of two Pareto sets, and allows for a relative comparison between the two sets. The function D is defined by $D(A,B) := S(A+B)$ - $S(B)$. The calculated value represents the volume that is weakly dominated by front A but not by front B. A value of $D(A,B) = 1$ means that the front B is entirely weakly dominated by A. A value of $D(A,B) = 0$ means that no points in front B are dominated by front A. This metric is not commutative. $D(A,B)$ does not necessarily equal $1\text{-}D(A,B)$, so both must be calculated for comparison.

Figure 3.6 (originally published in [108]) illustrates the concepts behind the S and D volume calculations. It considers two different Pareto fronts (represented in 2-D for visual clarity, although the metrics apply to n-D problems). For the maximization problem represented in the left image, front 1 and front 2 clearly

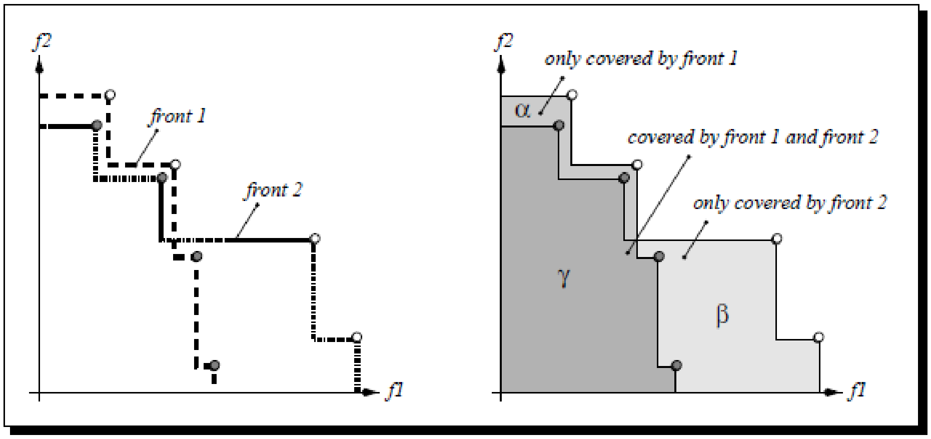


Figure 3.6: Illustration of the Hypervolume Quality Metric for Assessing Pareto Front Distribution. Figure was published in [108]

overlap, but neither is obviously better across both objectives. From the figure, *S(front 1)* is equal to the region that is the sum of α and γ and *S(front 2)* is equal to the region that is the sum of β and γ. The region of the figure covered by both front 1 and front 2 is *S(front1 + front 2)* $= \alpha + \beta + \gamma$. *D(front 1, front 2)*, then, is equal to α and *D(front 2, front 1)* is equal to β.

Zitzler has provided his C-language implementation of the hypervolume indicator as open source, and it has been used in this research [106]. This code is called with a specified number of objective functions and either one or two text files. The text files contain the Pareto data (where columns correspond to separate objective functions) for a given Pareto set. If only one data file is provided to the hypervolume code, it calculates only the S metric for the Pareto data. If two text files are provided, the hypervolume code outputs the S metrics for each of the data files, and also *D(data1, data2)* and *D(data2, data1)*.

However, it should be noted that to use this hypervolume code, the input data files must have the same number of objectives. In each of the hypervolume calculations in the following section, it will become evident how this fact was used. In essence, the objective numbers from the reduced objective set were used to identify which objectives from the control objective set to use. This is detailed further when the algorithm is used in Chapter 4.

3.2.5 Many-Objective Optimization Extension

It is not surprising that for large, complicated problems, the standard MOGA algorithms do not converge. It is well documented in optimization research that as the number of objective functions and constraints increases, it becomes more likely that progressive generations fail to improve the solution set by a significant degree. Brockhoff [11] explained that as the objective space is widened by an increasing number of objectives, the probability that design points are non-dominated increases. In other words, the probability that one design point is better across all objectives than another design point becomes statistically less likely. In this case, a larger number of Pareto solutions would be kept between generations to maintain the diversity. This would result in a lower selection pressure to find new solutions. In essence, the fitness between the early generations of a MOGA might not change significantly, and the algorithm may stop searching. According to Ishibuchi [46], the number of solutions required to approximate the Pareto set increases exponentially with the dimensionality (number of objective functions) of the problem.

There is a significant quantity of research in the literature devoted to alleviating these problems. The field is called Many-Objective Optimization (MaOO), and generally refers to problems with more than two objective functions. Ishibuchi [46] identifies five different categories of MaOO research: selection pressure, indicator-based search, preference based search, dimension/objective reduction, and visualization techniques.

There are many existing algorithms from journal papers that could be integrated into a MOGA (or other solution technique) to aid in algorithm convergence. These algorithms can be integrated to increase selection pressure (relax the Pareto dominance criterion [1], [77], [28]), change from a domination algorithm to an indicator-based search (use a different method besides domination for ranking [44], [109]), or use dimension and objective reduction ([79], [11], [70]). If applying one algorithm from this field does not work for a given problem domain, a different algorithms could be used to reach convergence.

Of these categories, objective reduction was selected to be utilized in this research. The final application domain used in this research will involve an objective function set consisting of 15 objectives - clearly in the realm of MaOO. This research's methodology proposes using an objective reduction algorithm to reduce the objective set prior to running Matlab's MOGA and to observe how well this improves convergence. Objective reduction has a straightforward conceptual analogue to the intended application of this research (HRT configuration selection through performance metric analysis), and analysis of the omitted objectives and remaining objectives will increase knowledge about the problem domain.

3.3 Offline Objective Reduction

Solving a problem with a large number of objectives can encounter several standard challenges. First, the large number of objectives makes a larger set of trade-offs between solutions. In other words, there is a large set of design variables that can be varied between solutions, and the size of the solution space increases. Second, a designer seeking to select a single solution from an optimized set has a much larger volume of traits to consider between the solutions. This can be impossible for a human mind to maintain in a logical way because of the difficulty in visualizing high-dimensional problems. Third, as the number of objective functions under consideration increases, the amount of computational time drastically increases.

Omitting objective functions from consideration has the potential to greatly improve computation time, and greatly simplify a problem. It is important to know quantitatively, however, how omitting an objective affects the problem characteristics. If the addition of an objective function to the set appears to have no effect on the resulting solution set, then the problem domain is insensitive to this objective. An objective that does not affect the solution does not contribute to the optimization of the problem and can be logically neglected from further analysis.

Alternatively, if including an objective function in the analysis greatly changes the solution set, then the problem is sensitive to this objective. Logically, this objective should be included in further analysis. In other words, as with any optimization problem, it would be expected that as the objective functions for a given problem are changed (either by weighting or by a different set of functions), the resulting solution set will be correspondingly changed.

In application, this concept will be used to specify a reduced set of objective functions to be applied iteratively with a specific mission scenario to observe the effect on the generated solution sets. This will provide both validation that the entire design space has been searched, and that the final solution set represents the best overall solutions.

A significant body of literature exists that is directly related to this problem of reducing the number of objective functions in a multi-objective optimization problem. It is sometimes called dimensionality reduction, and is a common problem in the fields of pattern recognition and image processing. In general, these methods can be divided into two realms: feature selection, and feature extraction. Feature selection involves downsizing from an existing set of objectives, creating a subset. Feature extraction draws new, non-redundant relationships and information from an existing body of data. Feature extraction can involve constructing combinations of variables. Both can be very useful in analyzing a large volume of data.

In the optimization literature, feature extraction is best represented as principle component analysis and feature selection by conflict-based approaches. Feature extraction allows synthesis of problem characteristics to form new relationships, while feature selection works with the existing problem characteristics. Both methods use different principles to preserve the properties of the underlying optimization problems. Both of these approaches can be used either *a posteriori* or in real-time during a search algorithm.

3.3.1 Selecting an Objective Reduction Method

Feature selection was utilized in this research because of the primary application domain (the HRT configuration selection problem). The set of relevant performance metrics of the HRT configuration selection problem resulting from objective reduction will be most useful if it clearly reflects the input objectives. Rather than coming up with a composite numeric answer that represents a combination of performance metrics, feature selection will allow easy traceability between selected and omitted objectives, and provide valuable information about the important problem components.

Objective reduction can be performed at two different places within a problem domain. In real-time (online objective reduction) often occurs within an iterative evolutionary algorithm, updating and reducing the objective function set between generations. Offline objective reduction is performed on an existing body of data, after a solution set has been found.

Offline objective reduction can specify which objectives are redundant within a set. In other words, offline objective reduction can determine which objectives do not contribute new information to the problem structure. It can also be used to illuminate various aspects of the problem domain to assist in *a posteriori* decision making and solution selection. Offline objective reduction is utilized in this methodology.

Three primary approaches to objective reduction have been greatly examined in the literature. Saxena and Deb [23, 78, 79] developed several online objective reduction methods based on principle component analysis, and have focused on incorporating constraint reduction into the multi-objective optimization. Jaimes and Coello Coello [54, 47] integrated conflict-based objective reduction into a multi-objective evolutionary algorithm by using correlation between nondominated vectors to estimate the conflict between each pair of objectives. Their approach scheduled the reduction stages during a MOEA to decrease the number of objectives that needed to be evaluated during the search. This work relied on several arbitrary variables that required an adaptive algorithm.

The third approach to objective reduction was deemed most valuable for this research. Brockhoff and Zitzler's conflict-based approach [10, 11, 12] to objective reduction sought to find the minimum objective set for a given optimization problem that preserves the underlying dominance structure. The conflict-based approach to objective reduction follows the feature selection framework and can be implemented both during a MOEA and *a posteriori*. Solution pairs are compared separately. Brockhoff developed both an exact algorithm and several heuristic algorithms for objective reduction.

The heuristic algorithms, based on a greedy algorithm approach, calculate a minimal objective subset that has at most a δ-error in the resulting dominance structure. It is possible to run both the exact algorithm and the heuristic algorithms for the case of δ-error $= 0\%$. The exact algorithm will find a smaller subset of objectives, but it will be overwhelmed with the computational complexity in the case of a larger number of objectives. The exact algorithm's run time is polynomial with respect to the size of the solution set, but is exponential in the number of objectives. The greedy heuristic will have a much smaller run time, with run time only becoming an issue for problems with more than 50 objectives and more than 200 solution pairs [12]. The main disadvantage of the greedy methods is that they are not guaranteed to find the minimum objective set, merely a greatly reduced set (minimal set). The resulting set size of the minimal objective set is not significantly larger with the heuristic algorithm than with the exact algorithm [12]. For the HRT configuration selection problem, this performance of the greedy heuristic algorithm was deemed sufficient.

For application in this dissertation, objective reduction will be used in two different types of applications: in the knapsack problem domain and in the human-robot team configuration selection domain, both as *a posteriori* analysis. It is highly unlikely that any of the objectives in either of these domains will be wholly redundant. In other words, it is anticipated that there will be some level of conflict between each pair of objectives, and some new information provided by each. Objective reduction will be used to find the different degrees of conflict. Brockhoff's greedy δ-MOSS algorithm was implemented for this research to facilitate work in these two domains.

The minimum size of an objective set will increase with the number of objectives up to a specific point, dependent on the number of solutions [10]. It is logical that the larger the search space, the more information that will be needed to characterize it, and therefore a larger minimum set size will be needed.

This research implemented Brockhoff's heuristic greedy δ-MOSS algorithm [12] to be used in *a posteriori* analysis of multi-objective optimization problems. It is anticipated that application to both the knapsack and HRT problem domains will reduce problem complexity and reduce the amount of information that needs to be looked at during the solution decision-making process.

3.3.2 Brief Overview of Critical Definitions and Concepts

There are several critical definitions and concepts that are necessary to understand the implementation of the objective reduction methodology used in this dissertation. These are reviewed briefly below, with definitions from Brockhoff. For a more detailed description of the definitions and their influence on the algorithms, please refer to [10, 12].

Fundamentally, the objective reduction algorithm compares all solution pairs across the objectives. One solution ($\vec{x}$) dominating another ($\vec{y}$) means that it is better in all objectives than the second. With each of the following definitions, the objective function set represented by F is notional: it can represent either a single objective function or a set of objective functions. A Pareto front resulting from a genetic algorithm is the set of points that are either non-dominated or incomparable, i.e. no solutions are better than the others in the set across all objectives. A strict definition of weak dominance is as follows:

Definition. *A solution $\vec{x}$* **weakly dominates** *a solution $\vec{y}$ if and only if $\vec{x}$ is not worse than $\vec{y}$ in all objectives in the set F, such that*

$$\vec{x} \leq_F \vec{y} :\Leftrightarrow \forall f_i \in F : f_i(\vec{x}) \leq f_i(\vec{y}) \tag{3.1}$$

Definition. *A solution $\vec{x}$ and a solution $\vec{y}$ are* **comparable** *if either $\vec{x} \leq_F \vec{y}$ or $\vec{y}$*

$\leq_F \vec{x}$.

$\vec{x}$ and $\vec{y}$ are **incomparable** *if neither $\vec{x} \leq_F \vec{y}$ nor $\vec{y} \leq_F \vec{x}$.*

A Pareto optimal solution set for an arbitrary optimization problem contains all solutions that either weakly dominate or are incomparable to any other solution.

In the conflict-based approach to objective reduction of the δ-MOSS algorithm, the underlying domination structure according to each objective function is determined (which points dominate each other according to each objective). A short-hand notation is adopted to represent the dominance relations of an objective function by the symbol $\leq_F$ for each different objective or set. If two different objectives, or two different objective subsets, do not have the same dominance relations, they are said to be conflicting.

Definition. *Two objective function sets F_1 and F_2 are called* **conflicting** *if their weak Pareto dominance relations differ, that is,*

$$\leq_{F_1} \neq \leq_{F_2}$$

. *The objective sets are called* **nonconflicting** *otherwise*

$$\leq_{F_1} = \leq_{F_2}$$

.

When attempting to reduce the complexity of a problem, it is often desirable to identify any redundant features or objective functions. These add no additional information. In the context of conflict-based analysis of objective functions, redundant objectives contain the same dominance relations.

Definition. *A set $F' \subseteq F$ of objectives is called* **redundant** *if and only if there exists an objective subset $F'' \subset F'$ that is nonconflicting with F'.*

In other words, if there is a subset of objectives that contains the same dominance relations as the whole set, then the redundant objectives can be omitted without altering the underlying dominance structure of the whole set.

In many practical applications, the dominance relations are too strict to allow for objective reduction. In other words, removing any objectives from the set would change the dominance structure of the problem. Instead, an acceptable error margin can be used to define a δ-nonconflicting reduced objective set.

Definition. *Based on the weak δ-dominance relation shown in the following equation, a* **δ-nonconflicting set** *has a maximum error of δ when it is wrongly assumed that $\vec{x} \leq_{F'} \vec{y}$. Then $\vec{x}$ is not worse than $\vec{y}$ in all objectives by the additive term δ.*

$$\leq^{\delta}_{F'} := (\vec{x}, \vec{y}) \mid \vec{x}, \vec{y} \in X \wedge \forall i \in F' : f_i(\vec{x}) - \delta \leq f_i(\vec{y}) \tag{3.2}$$

The term δ-nonconflicting can also be used to compare the dominance relations of two objective sets as follows.

Definition. *Let F_1 and F_2 be two objective sets. We call F_1* **δ-nonconflicting** *with F_2 if and only if both $(\leq_{F_1} \subseteq \leq^{\delta}_{F_2})$ and $(\leq_{F_2} \subseteq \leq^{\delta}_{F_1})$ holds; otherwise F_1 and F_2 are denoted as* **δ-conflicting**.

In the first case from the definition, the inclusion of the error margin means that F_1 and F_2 have become nonconflicting sets even when their strict dominance relations might have made them conflicting.

Definition. *Let δ_1, $\delta_2 \in \Re$ and F_1, F_2 be two objective subsets. The (δ_1,δ_2)-dominance relation*

$$\leq^{\delta_1,\delta_2}_{F1,F2}$$

on χ is defined as

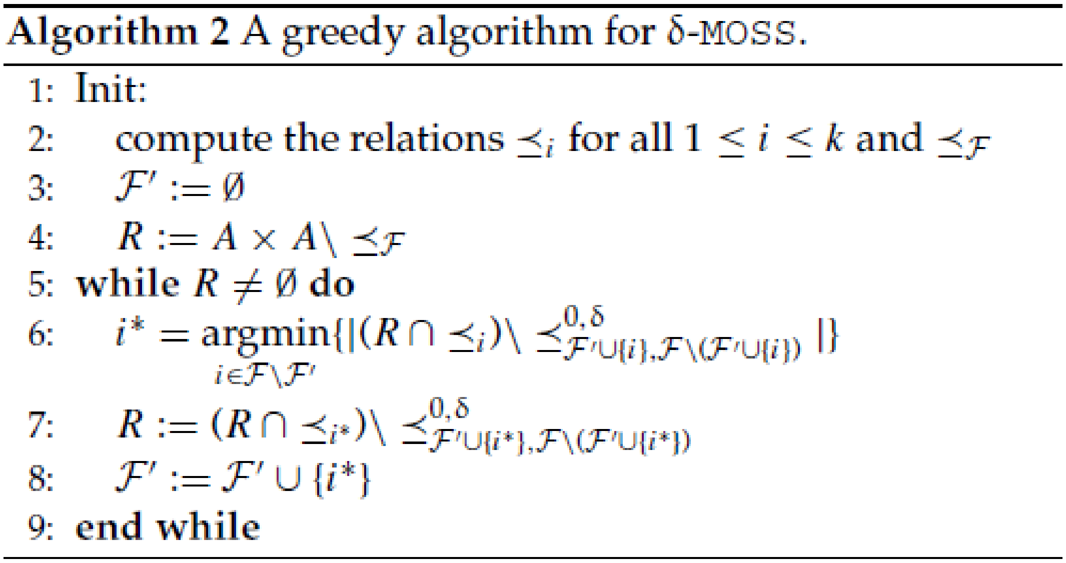
$$\vec{x} \leq^{\delta_1,\delta_2}_{F_1,F_2} \vec{y} :\Leftrightarrow F_1(\vec{x}) - \delta_1 \leq F_1(\vec{y}) \wedge F_2(\vec{x}) - \delta_2 \leq F_2(\vec{y}), \forall(\vec{x},\vec{y}) \in X. \tag{3.3}$$

where the $\wedge$ symbol represents a logical "and" operator. If both of the relations for F_1 and F_2 exist, then the point pair ($\vec{x}$, $\vec{y}$) satisfies the (δ_1,δ_2)-dominance relation.

3.3.3 The δ-MOSS Objective Reduction Algorithm

Brockhoff has defined three types of closely-related problems. In the minimum objective subset (MOSS) problem, the goal is to find a minimum objective set while preserving the dominance structure of the original problem. It is highly likely, however, that this approach could be too restrictive, resulting in no reduction of the problem complexity.

In the δ-minimum objective subset (δ-MOSS) problem, the goal is to compute a reduced objective set that is δ-nonconflicting with the original problem. The special case when $\delta = 0$ is the MOSS problem. Given a maximum-tolerable δ-error, this problem formulation will determine a reduced objective subset that has at most that δ-percentage error in the underlying dominance structure caused by omitting objectives.



Algorithm 2 A greedy algorithm for δ-MOSS.

```
Init:
    compute the relations $\preceq_i$ for all $1 \leq i \leq k$ and $\preceq_{\mathcal{F}}$
    $\mathcal{F}' := \emptyset$
    $R := A \times A \setminus \preceq_{\mathcal{F}}$
while $R \neq \emptyset$ do
    $i^* = \underset{i \in \mathcal{F} \setminus \mathcal{F}'}{\mathrm{argmin}} \{ |(R \cap \preceq_i) \setminus \preceq^{0,\delta}_{\mathcal{F}' \cup \{i\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i\})} | \}$
    $R := (R \cap \preceq_{i^*}) \setminus \preceq^{0,\delta}_{\mathcal{F}' \cup \{i^*\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})}$
    $\mathcal{F}' := \mathcal{F}' \cup \{i^*\}$
end while
```

Figure 3.7: Brockhoff's Greedy δ-MOSS Algorithm [12]

In the third problem type, the minimum objective subset of size k with minimum error (k-EMOSS), the goal is to find an objective subset that is δ-nonconflicting with the original problem but with at most k-objectives in the resulting subset. This specialization of the problem can be useful if a specific desired final set size is known *a priori.*

For this research, Brockhoff's heuristic greedy δ-MOSS algorithm (algorithm 2 from [12]) was implemented. The algorithm is reproduced in Figure 3.7.

As Brockhoff described [12], the influence of a δ-error on the size of the resulting objective set depends greatly on the problem itself. The impact of the objective reduction algorithm on a problem will vary greatly depending on properties of the problem itself. While this algorithm has been demonstrated to be a phenomenal tool, it might fail to yield a reduced set in some instances, based solely on properties of the multi-objective problem itself. This should be kept in mind for future application of the algorithm.

3.4 Summary

Referring back to Figure 3.2, the first two steps of the research process have been here described. The methodology that underlies the reduction in the performance metrics for the HRT problem was described in section 3.1.3 and the objective reduction method selected was discussed in 3.3.3.

After the methodology was verified in the knapsack problem domain (see Chapter 4), it was applied to the HRT configuration selection problem (see the experiments of Chapter 5 and Chapter 6). Reinserting the original application domain, a dataset of robotic agents and humans of various capabilities was used to run a sample case through the methodology. The resulting solution set was a pseudo-optimal assignment of which configuration would be best to perform the specified mission.

The methodology that was developed in this chapter represents a unique contribution to the field of HRT performance analysis. Given a set of agents and given a diverse set of potential metrics, this methodology allows a designer to rigorously choose a team configuration and task allocation within the team to satisfy one or more metrics in an optimal manner. A designer will not have to make *a priori* decisions about which are the critical metrics.

Four different algorithms were synthesized and incorporated through a MOGA to enable use of this research's methodology. Implementation and application of each of these algorithms has been done previously in the research. The unique contribution of this methodology is using the synthesis of the algorithms in the HRT domain to achieve superior and novel performance analysis and evaluation of results.

This type of methodology has not been done in the research field, and will be a valuable, unique contribution. Using the three orthogonal axes analogy, this research will be on the plane that is defined by the performance metrics and optimization techniques. This allows the substitution of the knapsack problem for the HRT configuration problem because the application domain will be used to feed data sets into metrics. In other words, the solution technique is not limited to a specific application domain. This methodology can be used for a much wider range of applications. Having a generic, objective methodology like this will be a necessary step to advancing the goal of using cooperative human and robot cooperative teams in future space activities.

Chapter 4

Objective Reduction Validation and Knapsack Application Experiment

4.1 Objective Reduction Algorithm Validation

The greedy δ-MOSS objective reduction algorithm described in section 3.3 was implemented in Matlab for this research. Before the objective reduction algorithm can be utilized, however, problem information must be available. Ideally, an initial Pareto front representing feasible solutions would be input into the algorithm. A Pareto front contains valuable information - correlations between variables, trends of each objective, and an evaluation of the trade-off between the objectives.

An example from Brockhoff's paper [12] was used to validate the algorithm's implementation. The example's data has been reproduced in Figure 4.1. In this example, there are four separate objective functions, all to be minimized, each with values for four variables. Without allowing for an error in the underlying dominance structure of the problem, no reduction of the objective set was possible for this example. When the problem data was input to the δ-MOSS algorithm with a tight δ-error of 0, it returned the same results as anticipated from Brockhoff's description - that no reduction of the objective set size was possible.

With a δ-error of 0.5, the algorithm returned a reduced objective set of F'

$= \{F_1,\ F_3,\ F_4\}$. In other words, the algorithm identified that there is at most an error of 0.5 in the dominance relations when the entire objective set is replaced by this subset. It should be recalled that the algorithm does not guarantee to find the minimal objective set, nor the optimal objective set. The optimal reduced objective set for this example (from inspection of the data, as described in [12]) is $F = \{F_2,\ F_4\}$ when a δ-error of 0.5 is tolerated. This smaller reduced objective set is the 0.5-δ-minimum with respect to the entire objective set.

The difference between the δ-minimal and the δ-minimum reduced objective sets stems from the algorithm's method of selecting amongst a set of equal-scoring (after considerable computation) objective functions to place in the reduced set. The algorithm selects the first objective function in the list of equal-scoring. This caused the placement of F_1 into the reduced set rather than F_2. On the next iteration loop within the algorithm, the fact that F_1 had already been selected influenced the rest of the decisions regarding the dominance relations. If the objective functions F_1 and F_2 were switched in order before being input into the algorithm, a different reduced objective set would result.

As part of the algorithm implementation, a subfunction was written to calculate the δ-error that would result from using a specified objective subset rather than the full objective set. This subfunction was used to complete the validation of the algorithm implementation. Inputting $F = \{F_2,\ F_4\}$ as the reduced objective set yielded the result that the δ-error would be 0.5. Similarly, inputting $F = \{F_1,\ F_4\}$ as the reduced set yielded the result that the maximum δ-error would be 2.5. Both of these results also match those described in Brockhoff's work [12].

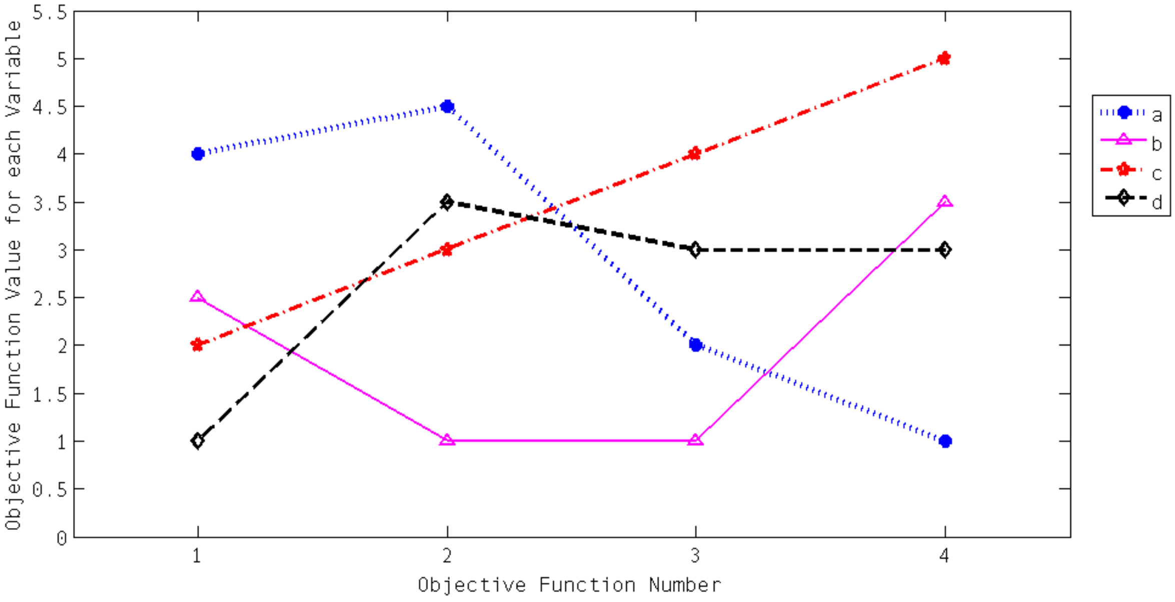


Figure 4.1: Objective Reduction Algorithm Validation Test Problem (reproduced from a similar figure in [12])

The goal of using this objective reduction algorithm is to reduce an unwieldy-sized full objective set to a more manageable version without changing the underlying dominance structure of the optimization problem. The δ-MOSS algorithm performs this task very well. It is not required to come up with the optimal solution to these complex over-constrained HRT-configuration selection problems. A reduced-complexity problem is the desired result.

4.2 Knapsack Test Problems Experimentation

The first set of experiments run in this dissertation were to demonstrate the utility of using the new methodology laid out in Chapter 3. It was anticipated that the δ-MOSS objective reduction algorithm would significantly reduce the complexity of the well-known multi-objective knapsack problem, depending on the error tolerated in the underlying dominance structure. Please see section 3.1.4 for a thorough description of the reasoning behind using the knapsack application to conceptually simplify the HRT configuration selection problem, and for a complete explanation of the goals and reasoning behind this experiment.

In brief summary, the multi-objective knapsack test problems allow a thorough application of the methodology developed in Chapter 3. As seen in Figure 3.4, an initial population was generated and input to the δ-MOSS objective reduction algorithm. The resulting reduced objective set (if reduction was possible) was used within one of the established solution techniques for a knapsack problem: a multi-objective genetic algorithm. Using the full set of objectives yielded the experimental control solution set, while using the reduced objective set yielded the experimental solution set. Analysis was performed on the two separate solution sets to determine how well the experimental method reduced the problem complexity while maintaining the important traits of the underlying problem.

4.2.1 Knapsack Simulation Setup and Initial Results

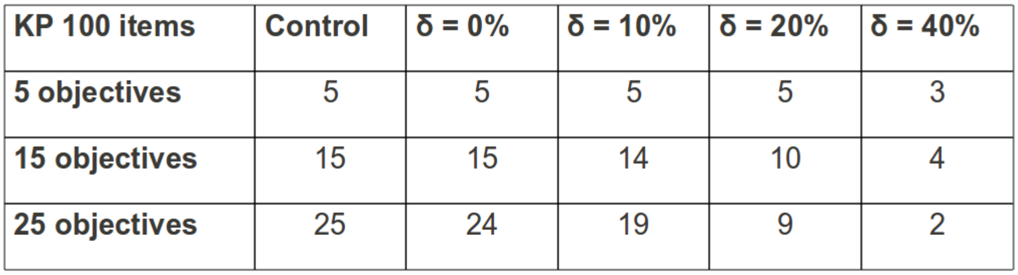
Brockhoff's seminal paper [12] used large-scale knapsack problem examples to demonstrate the power of the objective reduction algorithms. Some of his test problems were used for these simulations. The distinction between Brockhoff's analysis of the knapsack problems and the analysis that follows in this chapter, is that Brockhoff analyzed the reduction in the objective function sets for each of his test problems. In this research, the analysis was taken a step further to analyze the resulting Pareto solution sets to see how well the reduced sets modeled the underlying problem's dominance structure.

Three different knapsack test problem instances were used in this research effort. All three had 100 possible items for selection to place into the knapsack. Each problem instance contained profit values for these 100 items rated for 5, 10, and 15 objectives, respectively. The objectives within a knapsack problem were all represented as profit values. It was desired to have 100, 200, and 300 solutions, respectively, in the three problem instances.

The generic knapsack problem definition used as the basis of each problem instance differed between this implementation and Brockhoff's. This research used the version of the problem as defined by Zitzler [108], such that each item profit $p_{i,j}$ and each item weight $w_{i,j}$ were randomly chosen integers in the range of (10,100), inclusive. The capacity constraint was defined to be half of the sum of the item weights. This version of the knapsack problem was easily resizable for large-scale problem applications.

A script was written to follow the methodology laid out by Brockhoff. The script initially called a MOGA to evaluate a test problem instance and come up with an estimated Pareto front after 100 generations (with the corresponding required number of solutions in the Pareto front).

Brockhoff [12] recommended using δ-errors as relative percentages in more complex applications, in addition to using scaled objective functions, to remove order of magnitude over-simplification of the problem. This approach was used in the



KP 100 items	Control	δ = 0%	δ = 10%	δ = 20%	δ = 40%
5 objectives	5	5	5	5	3
15 objectives	15	15	14	10	4
25 objectives	25	24	19	9	2

Figure 4.2: Comparison of the Effect of δ-Error on the Resulting Reduced Objective Set Size Across Different Knapsack Problem Instances. The Numbers in the Table Correspond to the Number of Objectives in the δ-Minimal Set.

following experiments. To achieve this goal, the Pareto front for each problem instance was scaled such that the difference between the largest and smallest objective function value in the population would yield a δ-error equal to 1. This scaled Pareto front information was input to the δ-MOSS objective reduction algorithm.

Four different relative δ-error values were used: 0%, 10%, 20%, and 40% (the same as in [12]) to observe the effect of allowing errors in the underlying dominance relations on the resulting reduced objective set size and the quality of the resulting Pareto set. The results of the objective set size are tabulated in Figure 4.2.

The algorithm performed as expected. Due to the complex nature of these problems, a reduction in the objective function set size was not possible in the 5-objective and 15-objective problem instances without introducing error into the underlying dominance structure of the test problem. In the 5-objective test problem, a 40% error had to be introduced before reduction of the problem was possible.

Depending on the application of the objective reduction method, this level of error would be intolerable. This demonstrates one of the known limitations of the δ-MOSS algorithm - its ability to reduce the complexity of a problem is highly problem-dependent.

The 25-objective knapsack test problems demonstrated that they were much more conducive to objective reduction. With an increasing δ-error, fewer objectives were required to define the underlying dominance structure of each problem instance. This demonstrated similar performance of the δ-MOSS algorithm on the knapsack problem to that described in [12]. The value of this algorithm's performance should not be underestimated. Examination of Figure 4.2 reveals that the problem instance initially described by 25 objective functions was reduced to 19 objectives if a 10% error were acceptable, and down to 2 objectives if a 40% error were tolerable. While in practical applications a 40% error would be unlikely to be acceptable, allowing a 10% error could be entirely possible.

4.2.2 Solving the Knapsack Test Problems

Once the objective reduction algorithm had reduced the objective function set for each of the four error tolerances, each of the five test problems was input into the MOGA to solve the problem instance. Two different types of cases were computed here. In the control case, the full-size knapsack problems were run through the MOGA again. In the second case, the MOGA was run on each of the knapsack problems but with the four reduced objective sets found from the δ-MOSS algorithm.

All four δ values were run through the MOGA for each of the three test problems. For both of these cases (control and experiment), the MOGA was set to run 10 times, keeping the nondominated solutions across all of the runs.

When running the MOGA for this set of simulations, the built-in MOGA control options were altered. The MOGA options used previously had been intended to run through 100 generations and then stop the search process. The changes made for this set of experiments was to have the MOGA run through its entire search process and come up with its best Pareto front for the given problem. The MOGA was set to run for 400 generations in each of the problem instances, maintaining a population of 200 individuals, until either the maximum generation counter was reached or the average spread in the Pareto front was on the order of 10^{-6} (tolerance to determine that the Pareto front has converged).

4.2.2.1 Results From the 5-Objective Knapsack Problem Instances

Table 4.1 itemizes the initial results from each of the resulting final Pareto populations for the 5-objective knapsack test problems. Referring back to Figure 4.2, the 5-objective knapsack test problem required 5 objectives for all of the δ-error values except the 40% error test case, when the objective set size was reduced to 3 objectives.

It is most illuminating to compare the control column to the $\delta = 40\%$ column from Table 4.1. The same maximum number of items was found for both cases. The maximum total knapsack profit was slightly higher in the control case than in

KP 100 Items 5 Objs Problem Instance	Control	$\delta = 0\%$	$\delta = 10\%$	$\delta = 20\%$	$\delta = 40\%$
Final population size	791	814	783	805	401
Minimum # of items	50	51	50	51	51
Maximum # of items	64	64	64	63	64
Total profit maximum	4094	4070	4094	4012	3942
Profit minimum	2482	2576	2657	2580	2649
Average profit	3416	3421	3405	3465	3461
Weight maximum	3036	3036	3036	3036	3036
Weight minimum	2747	2727	2783	2804	2882
Average weight	2999	2999	3002	3006	3009
Weight constraint	3036	3036	3036	3036	3036

Table 4.1: Final Population Data for the 5 Objective Knapsack Problem Instances

the $\delta = 40\%$ case (3% higher final value in the control case), although the reverse was true for the minimum total knapsack profit. Even with including a 40% error into the underlying dominance structure, the best solution from the experimental MOGA run only differed from the control best solution by 3%. The experimental MOGA's worst solution (minimum profit value from the Pareto set) was 6.7% higher than the worst solution from the control case. From this first Pareto set analysis, it appeared that the experimental solution set well represented the control solution set even though it had been derived from a greatly reduced complexity problem description.

It is intriguing to note that all five test problems had (approximately) the same minimum and maximum number of items packed in the knapsack, and that they all found a way to pack the knapsack to reach the maximum weight constraint. The constraint handling algorithm appears to be working well.

While the data in Table 4.1 contains useful information about the resulting populations from each problem instance, it would also be helpful to analyze the spread and dispersion of the data points. Figure 4.3 displays the spread of solution set profit values for each of the five test problems as a boxplot. Each of the objective functions are displayed separately in the figure to aid in assessing how well the MOGA run optimized each individually. The x-axis for each of the subplots has been labeled with the objective function number that the data corresponds with. It should be noted that the ordering of the objectives in each of these plots corresponds with each objective function's selection by the δ-MOSS algorithm as being crucial to preserving the underlying dominance structure (left to right represents most to least important).

There are several points of interest that can be gleaned from Figure 4.3. For the δ-error values of 0%, 10%, and 20%, the algorithm determined that all five objectives were required to maintain the dominance structure. In other words, even with a 20% error allowance, removing an objective would change the dominance structure of the test problem by more than 20%. Additionally, for each of these three error values the δ-MOSS algorithm ordered the objective functions the same. Even with an increasing error tolerance up to 20%, the amount that each objective function contributed to the dominance structure remained the same.

This entirely changed with the $\delta = 40\%$ test problem instance. In this case,

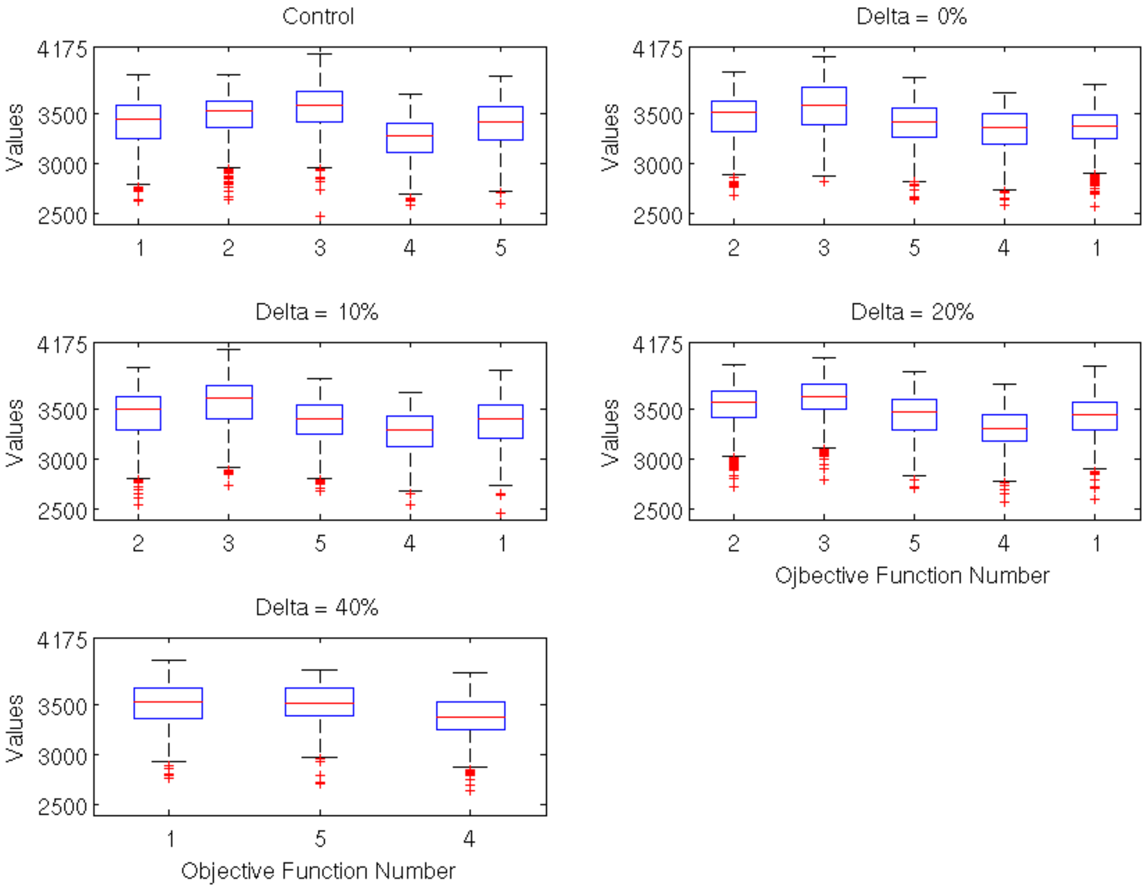


Figure 4.3: Final Pareto Population Data Resulting from Full MOGA Run for Each Knapsack Problem Instance with 5 Objectives

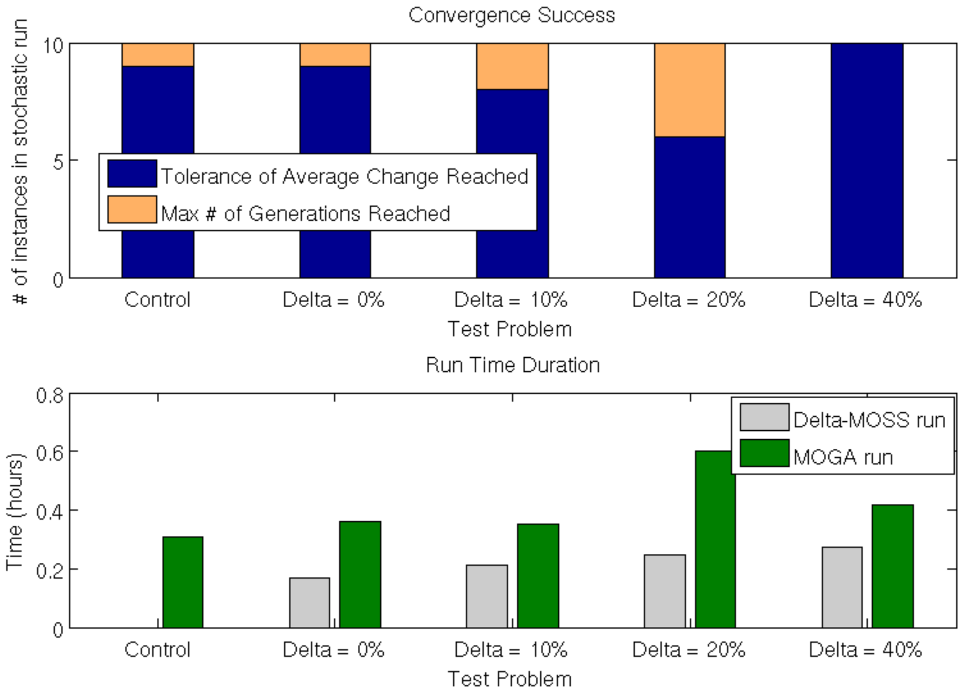


Figure 4.4: Termination Reason and Run Time for each Knapsack Problem Instance with 5 Objectives

the first objective was deemed most influential to the dominance structure, and was selected first. Allowing the large error removed the second and third objectives from consideration.

There are several low-value outliers for each of the objectives in all of the test cases. This means that not all of the solutions to the knapsack problem have pseudo-optimized solutions even after the MOGA options had been altered to facilitate convergence. This indicated that analysis was needed to assess the reason that each test problem's MOGA runs was terminated. Figure 4.4 concisely displays this data.

At the end of every MOGA run, the reason for the termination of that run had been stored. It had been desired that each run would only terminate when the tolerance of the average change in the Pareto spread was reached, indicating that the solution set had converged to a final set. As can be seen in Figure 4.4, this was the case for the majority of the MOGA runs across all five test problems.

However, a second termination reason turned up in all but the $\delta = 40\%$ case that indicated that not all of the MOGA runs reached convergence (the upper portions of the bar plot). When the pre-specified maximum number of generations had been reached (400 in this case), the MOGA stopped searching for solution convergence and ended the run with the current population. For both the control and the $\delta = 0\%$ case, this occurred once in the ten stochastic runs. In the $\delta = 10\%$ case, two out of the ten MOGA runs failed to converge. In the $\delta = 20\%$ case, four out of the ten MOGA runs failed to converge. All ten stochastic runs of the $\delta = 40\%$ test problem reached convergence. This is the anticipated result - with the problem complexity reduced from five objectives down to three objectives, it was easier for the MOGA to reach convergence.

The convergence results from the first four cases seem counter-intuitive. It would be anticipated that the larger the set of objective functions, the more difficult it would be for the MOGA to reach convergence and the more likely the runs would be to hit the maximum number of generations first. In this 5-objective test problem, however, each of the first four test problems had the same number of objectives being tested.

The reason for this divergence in termination reason can be thought of as the probability of flipping heads or tails with a coin. Each flip has a 50% chance of turning up heads, yet it is entirely possible to flip the coin 5 or 10 times and get tails every time. The small sampling of the dataset creates the appearance that tails has a greater than 50% chance of turning up. However, if a much larger dataset were run (if the coin was flipped 100 times), it becomes much more likely that tails will turn up only 50% of the time.

Further analysis was run on the 5-objective knapsack test problems to determine whether the transient probability behavior was exhibited in the convergence success of the MOGA. Figure 4.5 displays the results from this analysis. Instead of running the MOGA 10 times, the MOGA was run a total of 30 times, collating the non-dominated solutions from each.

In the control case and the $\delta = 0\%$ test case in Figure 4.5, it can be seen that Pareto front convergence was reached 66% of the time. A full third of the stochastic 30 runs failed to reach Pareto convergence. The other two test cases with 5 objectives (10% and 20% δ-error) had slightly better average convergence success, but still not as high as expected. In fact, this convergence rate was worse than that exhibited with the 10 MOGA runs.

It is also concerning that this convergence rate was achieved with the 5-objective knapsack problems because these were the least complex of the problems to be analyzed. This convergence failure can be attributed to two different factors. On the one hand, the objectives for the knapsack problem were based on randomly selected integer values. Not only were the distinct objectives not seeking concurrent goals (nor, for that matter, contradictory goals), there might not have been any order within them whatsoever. Additionally, a known challenge of MaOO is reaching solution convergence. There are just so many contradictory objectives and variables

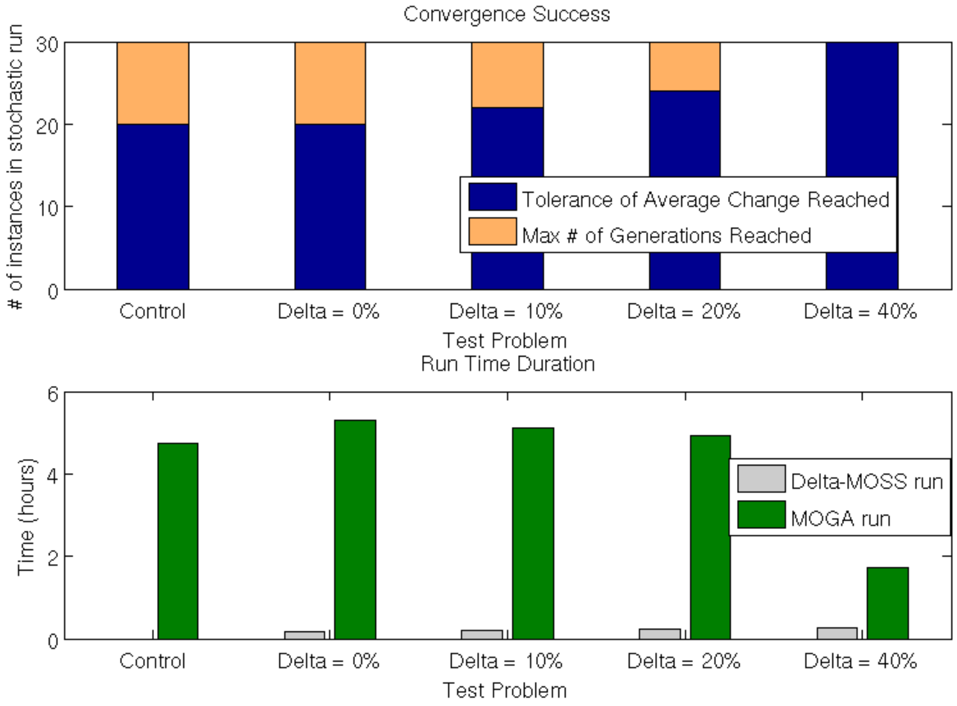


Figure 4.5: Termination Reason and Run Time for the 5-Objective Knapsack Cases, with 30 MOGA Runs

that convergence can become difficult. This convergence rate was noted for further comparison with the more complex test problems.

The second plot displayed in Figure 4.4 indicates the run time required for the two different components of this experiment: the run time of the greedy δ-MOSS objective reduction algorithm and the total run time of the 10 stochastic MOGA iterations.

It would be anticipated that the run time of the MOGA with the entire objective function set would be considerably longer than the time required to run the reduced objective function set, and that the resulting Pareto fronts would not be significantly different. This does not appear to be the case. The run time for the

MOGA stayed roughly the same for the $\delta = 0\%$ and 10% cases (this was the anticipated result because there was no difference in the number of objectives between these two cases and the control), but the $\delta = 20\%$ case (which also had five objectives) took twice as long as the control. This reflects the fact that roughly 25% of the MOGA runs in the $\delta = 20\%$ test case did not converge. Instead, these MOGA runs churned until the maximum number of generations was reached, greatly increasing the duration of those MOGA runs.

For the 5-objective problem instances, the run time for the greedy objective reduction algorithm increased slightly as the error-tolerance was increased. A reason for this is that with the error tolerance, more of the dominance relations overlapped resulting in a larger number of combinations to be rechecked.

4.2.2.2 Results from the 15-Objective Knapsack Problem Instances

The results for the 15-objective knapsack problem followed similar patterns to those obtained from the 5-objective function knapsack test problems. Table 4.2 itemizes the initial results for the 15-objective knapsack test problems. Referring back to Figure 4.2, the 15-objective knapsack test problems required all 15 objectives to maintain the underlying dominance structure of the problem if no δ-error was allowed. If a 10% error in the dominance structure was feasible, one objective function could be removed from the objective function set. All of the other 14 objectives were required to maintain problem detail.

Allowing a 20% δ-error in the dominance structure of the 15-objective knapsack test problem allowed a reduction in objective set size to 10 objectives, significantly reducing the problem complexity. As can be seen in Table 4.2, introducing a 40% δ-error reduced the objective set to 4 objectives. It is possible that this substantial reduction of the size of the objective set would cause considerable alteration to problem detail. A detailed examination of the solutions in the Pareto front for this test problem (as seen in Figure 4.6 and discussed below) revealed that this was not the case.

For the knapsack problem instances with 15 objectives, it is illuminating to compare across all five of the test problems for analysis of the initial results. Comparing the results tabulated in Table 4.2, several trends become apparent. All five test problems found a maximum total profit value within 2% of each other. In addition, reducing the objective set size resulted in an increase in the average solution profit value. This would lead to the conclusion that even the 40% δ-error Pareto set provided a good representation of the original 15-objective knapsack problem. From this initial Pareto set analysis, it appeared that the all of the experimental solution sets well represented the control solution set even though they had been derived from a greatly reduced complexity problem description.

While the data in Table 4.2 contains useful information about the resulting populations from each problem instance, it would also be helpful to analyze the spread and dispersion of the data points. Figure 4.6 displays the spread of solution set profit values for each of the five test problems in the 15-objective knapsack problem instance as a boxplot. The x-axis for each of the subplots has been labeled with the objective function number that the data corresponds with, and rearranged to re-

KP 100 Items 15 Objs	Control	$\delta = 0\%$	$\delta = 10\%$	$\delta = 20\%$	$\delta = 40\%$
Problem Instance					
Final population size	857	924	900	826	535
Minimum # of items	49	49	49	49	51
Maximum # of items	64	65	65	65	65
Total profit maximum	4010	4077	4051	4097	4086
Profit minimum	2289	2223	2098	2400	2442
Average profit	3350	3365	3362	3369	3398
Weight maximum	2685	2685	2685	2685	2685
Weight minimum	2431	2485	2480	2451	2448
Average weight	2650	2652	2651	2655	2656
Weight constraint	2685	2685	2685	2685	2685

Table 4.2: Final Population Data for the 15-Objective Knapsack Problem Instances

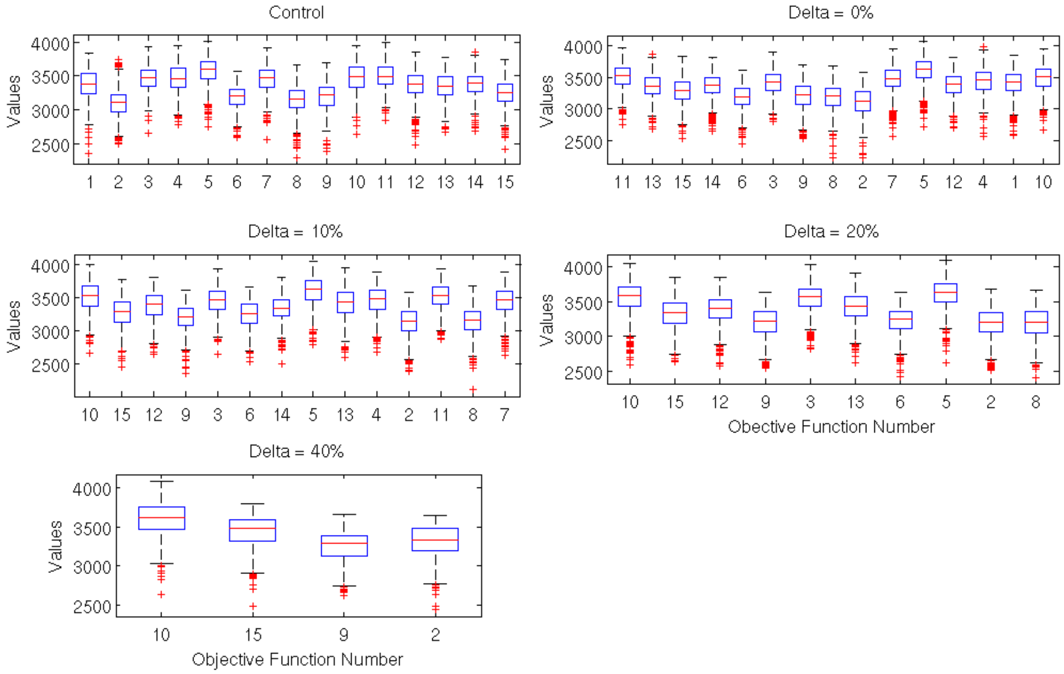


Figure 4.6: Final Pareto Population Data Resulting from Full MOGA Run for Each Knapsack Problem Instance with 15 Objectives

flect the order of importance of each objective function in preserving the underlying dominance structure of the 15-objective problem.

There are several interesting trends to note from these boxplots. As had been the case with the 5-objective knapsack problem instance, the importance ordering of the objective functions resulting from the δ-MOSS algorithm changed significantly depending on the level of error allowed within the underlying dominance structure. In this case, the 10th objective was rated to be least important by the δ-MOSS algorithm when restricted to 0% error, yet this same objective jumped to most important once an error was introduced (in the 10%, 20% and 40% cases). This is the most extreme example of the objective function reordering. It provides insight into how dependent the selection of objective functions was to the error value.

Another trend that was apparent in the 5-objective knapsack problem instance and is evident here is that there were several low-value outliers for each of the objectives in all of the test cases. This means that not all of the solutions found for the knapsack problem were pseudo-optimized. In other words, it is likely that these outliers were not on the Pareto front and, with further generations of the MOGA, could have been improved.

Figure 4.7 displays that the MOGA toolbox struggled to converge the 15-objective knapsack test problem instance. In the control case, less than half of the stochastic 10 runs through the MOGA reached convergence - instead, those runs were automatically stopped after 400 generations had failed to converge. It would be intuitive that with the decreased objective set size of the other cases in this problem instance, that Matlab's MOGA would have a higher convergence success rate. The $\delta = 10\%$ test case had one fewer objective function, and had a 20% higher success rate at reaching convergence.

This trend, however, was not seen in the $\delta = 20\%$ test case. With 5 fewer objectives than the control, it would be intuitive to assume that it would have a greater convergence success. As seen in Figure 4.7, this was not the case. Only 2 out of the 10 stochastic MOGA runs converged.

The second plot in Figure 4.7 collated the run time for the 5 test cases for both the δ-MOSS algorithm and the run time for 10 MOGA runs with the reduced objective function sets. It is interesting to note that while the MOGA run time for the control, $\delta = 0\%$, and $\delta = 10\%$ cases remained relatively constant, there was both an increase in run time for the $\delta = 20\%$ test case and a substantial decrease in run time for the $\delta = 40\%$ case, the former an initially non-intuitive result and the latter more aligned with predictions. The reason for the longer run time in the $\delta = 20\%$ test case was the low convergence rate of the MOGA runs. Rather than reaching convergence and moving on to the next run, each of the 10 MOGA runs iterated through the full 400 generations.

Another feature worth noting from the time plot in Figure 4.7 was that the run time for the δ-MOSS algorithm appeared to increase with each subsequent increase in the allowed error in the dominance structure. The δ-MOSS algorithm performs a pairwise comparison of each of the Pareto solutions across all of the objective functions. With an increased error tolerance, more of the pairwise relations could move in either direction - the dominance relations became much more fluid. This required a substantial increase in the number of pairs that needed to be compared.

4.2.2.3 Results from the 25-Objective Knapsack Problem Instances

The results for the 25-objective knapsack problem follow similar patterns to those obtained from the 5 and 15-objective function knapsack test problems. Table 4.3 itemizes the initial results for the 25 objective knapsack test problem. Referring back to Figure 4.2, the δ-MOSS algorithm indicated that one of the objectives from the 25-objective knapsack test problem could be removed with 0% error in the underlying dominance structure of the problem. In other words, one of the 25 objectives was fully redundant and provided no additional information to the

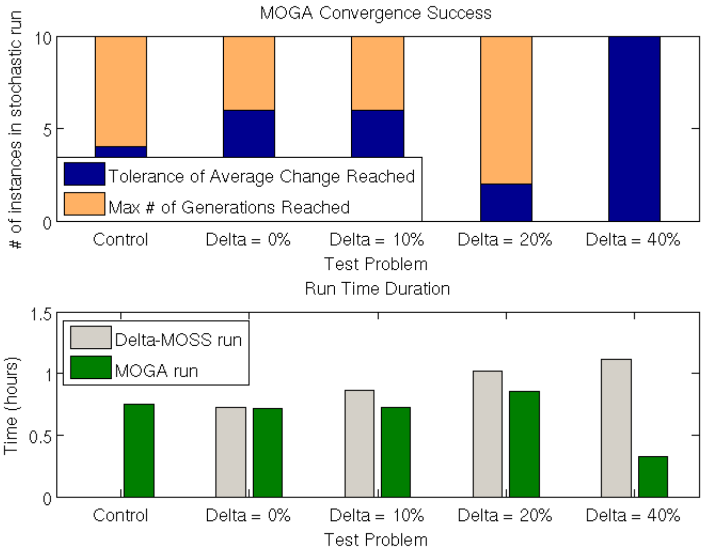


Figure 4.7: Run Time and Termination Reason for each Knapsack Problem Instance with 15 Objectives

problem. If a 10% error in the dominance structure were feasible, five objective functions could be removed from the objective function set - a substantial reduction in problem complexity facilitated by a small induced error.

Allowing a 20% δ-error in the dominance structure of the 25-objective knapsack test problem resulted in reducing the objective set size by more than half. In other words, allowing a 20% error effectively turned 15 of the objectives into redundant objectives that provided no new information about the underlying problem. Introducing an error of 40% reduced the problem to a two objective test case, effectively making 23 of the objectives contain wholly redundant information about the dominance structure.

The significant reduction in problem complexity caused by allowing a 20% and a 40% δ-error should lead to questions regarding the validity of the resulting experimental Pareto solution sets. These questions will be answered with Figure 4.8 and its discussion.

For an initial analysis of the quality of the solution sets from each of the five test cases for the 25-objective knapsack problem, it is illuminating to compare the Pareto set results tabulated in Table 4.3.

The difference between the maximum total profit values for the 25-objective knapsack problem instances was twice what it had been in the 5 and 15-objective knapsack problem instances: a difference of 6% between the maximum profit derived from the control solution set and that from the 40% error solution set. However, the reduced objective set size for the 40% test case was already a cause for concern about its validity in replicating the original test problem. When the 40% error test case was removed from consideration, the percentage difference of the maxima profit values was within 3% of each other, the same trend observed with the 5 and 15-objective knapsack problem instances. In reducing the objective set down to two objectives in the 40% error test case, it appeared that the ability of the MOGA to reach the maximum profit value of the control problem was removed.

The trends for the average solution profit value are more intriguing. Reducing the objective set size by allowing a 10% error raised the average solution profit by 1%. Increasing that error to 40% raised the average solution profit by 8%. This would lead to the conclusion that even though incorporating a 40% error value into the dominance structures of the 25-objective knapsack problem resulted in a lower maximum profit value for the solution set, the average profit value of the solution set had increased by 8%. In other words, the 40% error solution set did not find the

KP 100 Items 25 Objs Problem Instance	Control	$\delta = 0\%$	$\delta = 10\%$	$\delta = 20\%$	$\delta = 40\%$
Final population size	935	956	943	819	820
Minimum # of items	50	47	50	51	56
Maximum # of items	66	66	66	66	64
Total profit maximum	4165	4138	4110	4036	3920
Profit minimum	2214	2142	2309	2392	3026
Average profit	3351	3333	3390	3433	3614
Weight maximum	2766	2766	2766	2766	2766
Weight minimum	2414	2483	2543	2535	2624
Average weight	2728	2731	2732	2740	2739
Weight constraint	2766	2766	2766	2766	2766

Table 4.3: Final Population Data for the 25-Objective Knapsack Problem Instances

extrema point that had been found by the other test cases, but it appears that the average overall solution had a higher profit. With the significantly reduced problem complexity represented by the reduced objective set of the 40% error test case, it could be argued that this Pareto set provided a good representation of the original 25-objective knapsack problem.

While the data in Table 4.3 contains useful information about the resulting populations from each problem instance, it would also be helpful to analyze the spread and distribution of the data points, as had been done in the previous knapsack problem instances. Figure 4.8 displays the spread of solution set profit values for each of the five test problems as a boxplot. The x-axis for each of the subplots has

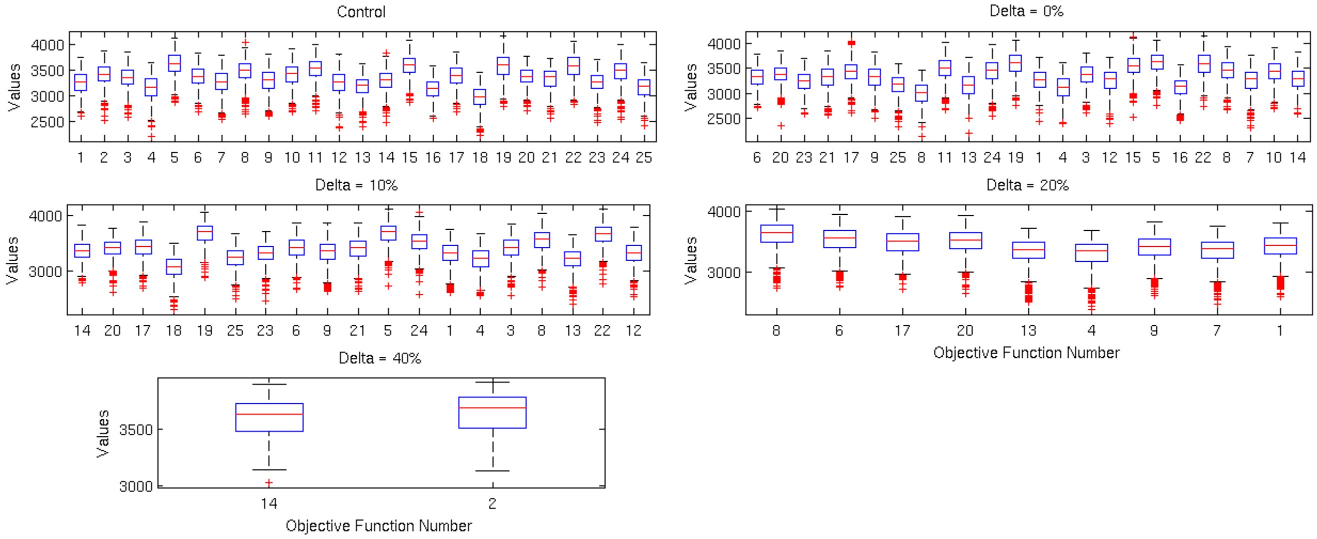


Figure 4.8: Final Pareto Population Data Resulting from Full MOGA Run for Each Knapsack Problem Instance with 25 Objectives

been labeled with the objective function number that the data corresponds with, and rearranged to reflect the order of importance of each objective function in preserving the underlying dominance structure of the 25-objective problem.

There are several interesting trends to note from the boxplots in Figure 4.8. As had been the case with the two previous knapsack problem instances (5 and 15 objective), the importance ordering of the objective functions resulting from the δ-MOSS algorithm changed significantly depending on the level of error allowed within the underlying dominance structure. In this case, there appeared to be much more fluidity in the importance of each objective depending on the error level. The 14th objective was deemed least important in the $\delta = 0\%$ test case, jumped to the most important in the $\delta = 10\%$ test case, was found to be wholly redundant in the $\delta = 20\%$ test case and was not even present in the reduced objective set, yet the same objective jumped back to the most important slot in the $\delta = 40\%$ test case.

The same fluidity of objective function importance can be seen with the second objective function. It was deemed wholly redundant and was not present in the reduced objective sets of the $\delta = 0\%$, 10%, and 20% test cases, yet was the second most important objective in the $\delta = 40\%$ test case. This was the most extreme example of the objective function reordering in the 25 objective knapsack problem instance. Combined with the transience of the 14th objective, it provides insight into how dependent the selection of objective functions was to the error value and to the nature of the underlying problem.

As with the two previous knapsack problem instances (5 and 15 objective), there were several low-value outliers for each of the objectives in all but the last of the test cases. This is unsurprising in the 25-objective test case. As with the other problem instances, this means that not all of the solutions found for the knapsack problem were pseudo-optimized. In other words, the MOGA runs did not reach full convergence.

Figure 4.9 displays that the MOGA toolbox struggled to converge the 25-objective knapsack test problem instance, although it appeared to have more success than with the 15-objective test cases. In the control case, 7 out of the 10 stochastic MOGA runs converged to a Pareto front. This was a very surprising result, considering how the MOGA struggled with the 15 objective. It can be assumed that the objectives in the 25-objective test problem were less in conflict than the 15-objective test case. The two were entirely separate problem instances with no overlap, both highly dependent on randomized objective values.

Equally surprising was the convergence failure of the $\delta = 20\%$ test case. Zero of the 10 stochastic MOGA runs reached convergence. All ran the full 400 possible generations without converging. This fact was also evident in the time plot from Figure 4.9 for the 20% case. The MOGA run time was greater for this test case than for any of the other test cases. It appears that the MOGA runs failed to find the extrema points for this test case, although the average solution values closely resembled those found in the other problem test cases.

In the 25-objective knapsack problem instance, the contrast between MOGA run time and δ-MOSS algorithm run time became glaringly obvious. For the greater complexity problem, the δ-MOSS algorithm took on average twice the stochastic MOGA run time (even the control MOGA case with 25 objectives) to reduce the objective set. If computation time is a limiting factor for a set of analysis, it appears that running the methodology proposed in this dissertation research (the δ-MOSS algorithm before the MOGA) is not only not beneficial, but costly.

It should be remembered, however, that the decision space that a designer has at the end of the methodology had significantly reduced complexity, facilitating the designer's selection of a final candidate solution. The following section applies a Pareto set quality metric to determine which of the 5 test cases for each of the 3 knapsack test problem instances resulted in a better decision space for the designer.

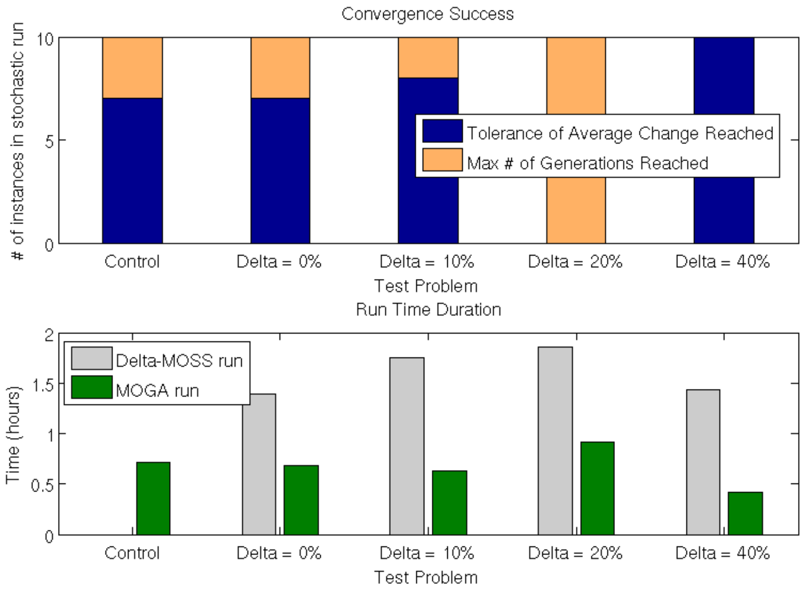


Figure 4.9: Run Time and Termination Reason for each Knapsack Problem Instance with 25 Objectives

4.2.3 Hypervolume Quality Metric Results for the Knapsack Problem

Figure 4.10 shows the results of applying the hypervolume indicator to the Pareto set resulting from the 5-objective knapsack problem instance. The requirement of the Pareto sets being compared to have the same number of objectives resulted in assessing the control Pareto set only across the objectives it had in common with each of the reduced objective sets. The four test cases each have four different hypervolume quality metric values displayed.

Examining first the S metric, the absolute Pareto volume coverage was calculated to be similar for the $\delta = 0\%$, 10% and 20% cases. Any discrepancy in the values for these cases suggests that the ordering of the objectives themselves when input into the MOGA aided Pareto convergence. This was the anticipated result

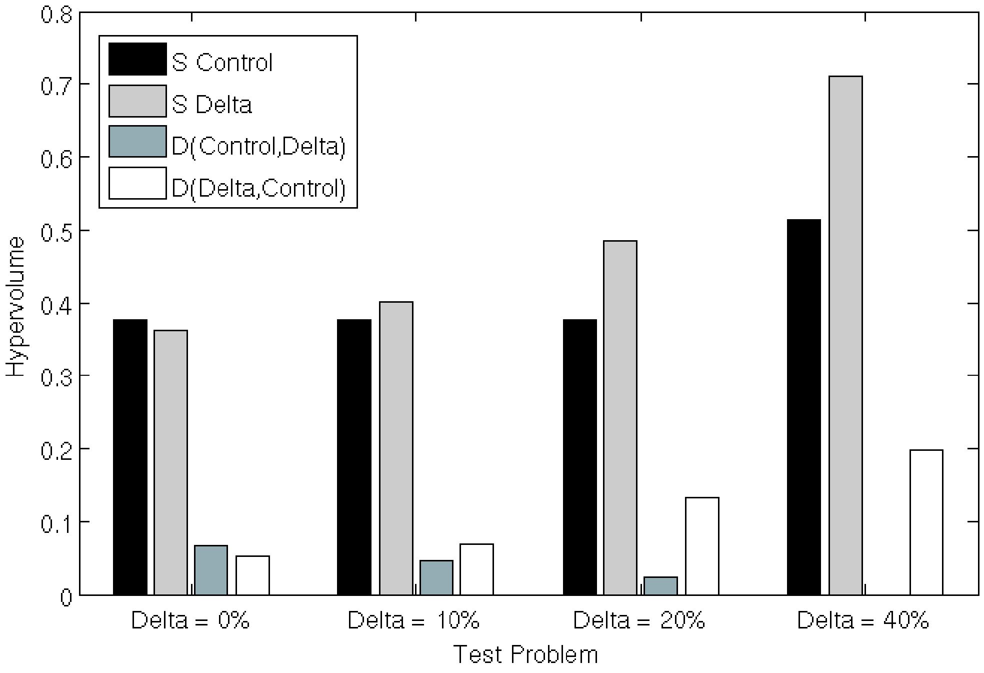


Figure 4.10: Hypervolume Quality Metric Calculations for Knapsack Problem Instance with 5 Objectives)

because in the 5-objective knapsack problem instance, the only test case that had a reduced objective set was the 40% error case. In the $\delta = 40\%$ test case, the S metric value was substantially higher than the control's. This indicates that the reduced objective set's Pareto set had better absolute coverage of the design space than the control Pareto set did.

Examination of the D metric clearly highlights the advantage of this dissertation's methodology in finding a better design space representation for the mission designer. In the 40% δ-error test case (the only one in this problem instance that truly reflects the reduction methodology), the value for *D(Control, Delta)* was zero. In other words, the control Pareto set had zero volume coverage that the δ Pareto set did not have. This by itself states that the Pareto set resulting from the reduced

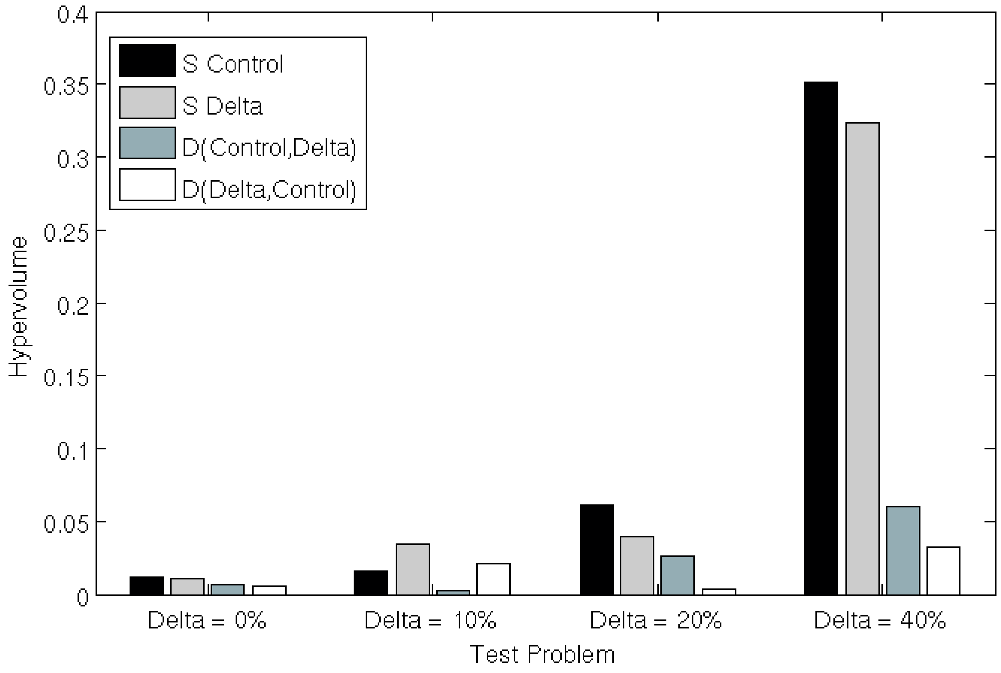


Figure 4.11: Hypervolume Quality Metric Calculations for Knapsack Problem Instance with 15 Objectives)

objective set had at least an equivalent volume coverage as the control case. Looking at the value for *D(Delta, Control)*, however, it becomes apparent that the Pareto set resulting from the $\delta = 40\%$ error case actually had 20% better volume coverage than the control Pareto.

The results from these quality metrics reveal that the solution set derived from application of this research's methodology had an overall better demonstrated representation of the Pareto front. The diversity of the solutions and the coverage of the front itself was better. If computational time is not an issue in a designer's *a priori* analysis, implementation of this research's methodology resulted in an across-the-board improvement in the final solutions to the 5-objective knapsack problem instance.

Figure 4.11 displays the hypervolume indicator quality metric results for the 15-objective knapsack problem instance. Recall that the reduced objective set sizes for the four test cases were 15, 14, 10, and 4, respectively. The $\delta = 40\%$ test case is most visible in this Figure. The S metric value from the control Pareto set was 3% better than that from the δ test case. Additionally, the relative volume coverage metric, *D(control, delta)*, was approximately 3% better than the *D(delta, control)* value. In other words, in both the S and D metric, the control Pareto set had better coverage of solutions than the δ case.

This result should be put in perspective. The reduced objective set for the $\delta = 40\%$ test case (which had 11 fewer objective functions than the control case) resulted in a Pareto solution set that was only 3% worse according to the quality metrics. That result is has great consequence. A drastic decrease in problem complexity was afforded at only a 3% decrease in solution quality. Considering the ease with which a mission designer could examine the δ's 4 objective reduced objective set and its Pareto solutions, it appears that, for this test case, this research's methodology proved to be very beneficial.

Figure 4.12 displays the same data as Figure 4.11 except without the $\delta = 40\%$ test case to allow for more granular detail to be visible. The same trends can be seen in this plot. Even though the control Pareto set's quality metric values for the $\delta = 20\%$ test case were better than those derived from the δ's Pareto set, the difference in solution quality was only 2%. Removing 5 objectives and their corresponding complexity from the problem description only resulted in a 2% difference in solution quality.

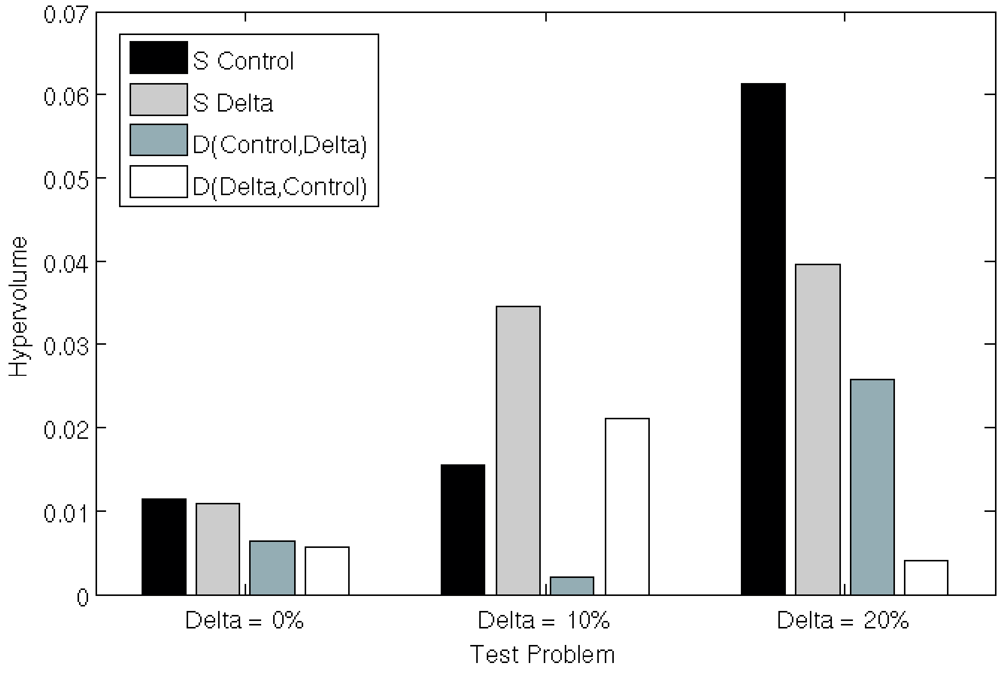


Figure 4.12: Enlarged Figure of Hypervolume Quality Metric Calculations for Knapsack Problem Instance with 15 Objectives)

The quality metric values shown in Figure 4.13 contain the results from the 25-objective knapsack problem instance. Recall that the reduced objective set sizes for the 25-objective knapsack problem instance for the $\delta = 0\%$, 10%, 20%, and 40% test cases were 24, 19, 9, and 2, respectively. The data in Figure 4.13 is even more persuasive for the utility of this research's methodology. The quality of the $\delta = 20\%$ Pareto solution set is approximately equivalent to the quality of the control solution set even though it represented a decrease in problem complexity of reducing the 25-objective control problem down to 9 objectives. To decrease complexity by nearly 70% and still maintain the same solution quality proves the merit of this research's methodology.

The $\delta = 40\%$ test case contains an even larger data contrast. The Pareto

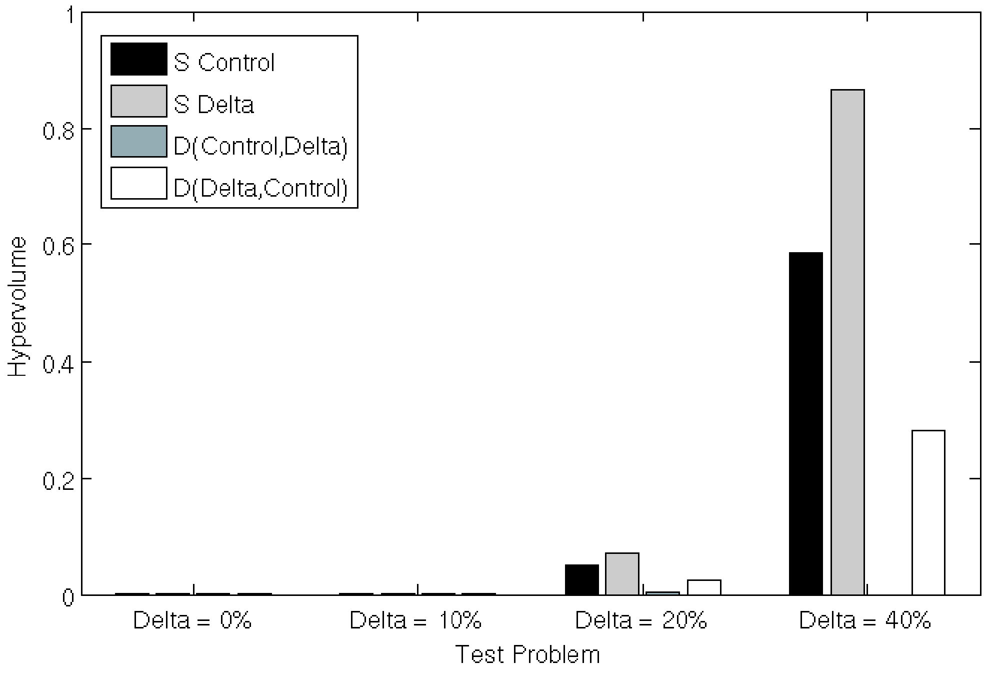


Figure 4.13: Hypervolume Quality Metric Calculations for Knapsack Problem Instance with 25 Objectives)

solution set resulting from the reduced objective set had nearly 30% better quality than the control! This suggests that the MOGA was unable to converge the two objectives represented in this control data set because it had so many other conflicting objectives to be concerned with. Using the reduced objective set, however, resulted in a much more optimized Pareto set.

For completeness, Figure 4.14 contains a close-up view of the $\delta = 0\%$ and 10% test cases for the 25 objective knapsack problem instance. Note the y-axis scale in this figure: 10^{-3}. The Pareto solution sets resulting from these error tolerances were approximately equivalent in quality to the control for each test case.

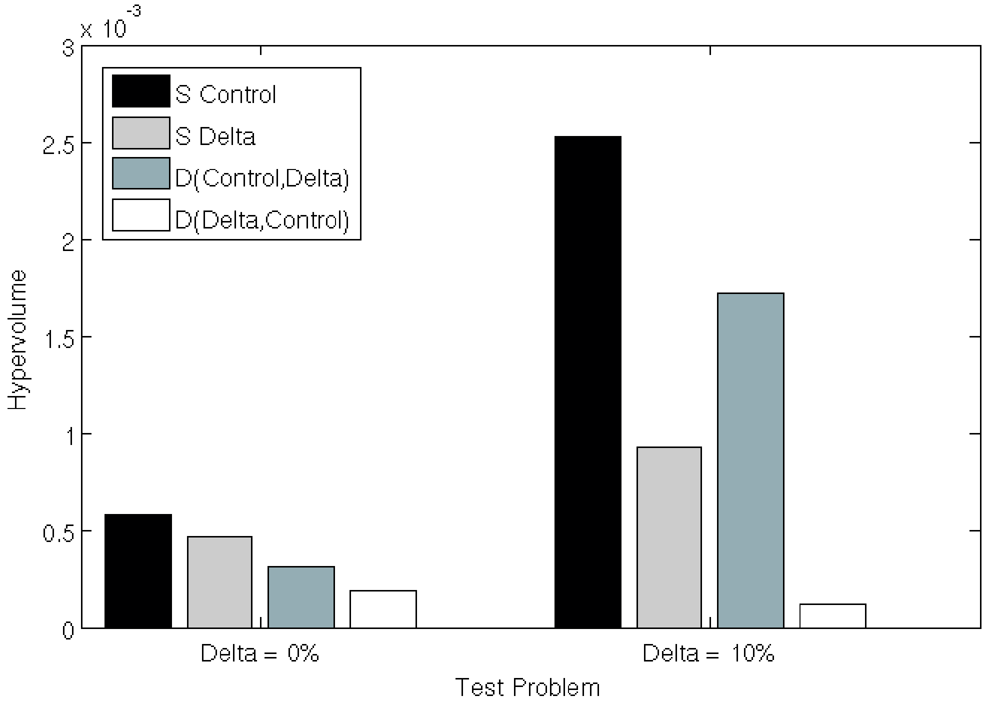


Figure 4.14: Enlarged Figure of Hypervolume Quality Metric Calculations for Knapsack Problem Instance with 25 Objectives)

4.3 Summary

Using the conceptually simpler knapsack problem domain, the utility of this dissertation's methodology has been demonstrated. The methodology does not prove its worth across the board. One very clear disadvantage to the methodology is the (potentially substantial) increase in computation time required to reach a Pareto solution for a given design problem. As seen from the 25-objective knapsack problem instance, if computation time is a limiting factor for a set of analysis, using this dissertation's proposed methodology would be a hindrance to the analysis effort.

However, if computation time is not an issue, the merit of this research's proposed methodology has been clearly demonstrated. If a mission designer is not concerned with producing results as fast as possible, the Pareto solution sets resulting from this methodology have been demonstrated to have been at least equivalent, if not better, than the control case for the knapsack problem application.

Recall the reduced objective set for the $\delta = 40\%$ test case from the 15-objective knapsack problem instance (which had 11 fewer objective functions than the control case) resulted in a Pareto solution set that was only 3% worse according to the quality metrics. A drastic decrease in problem complexity was afforded at only a 3% decrease in solution quality. Considering the ease with which a mission designer could examine the 4-objective reduced set, this research's methodology proved to be very beneficial.

The results from these quality metrics reveal that the solution set derived from application of this research's methodology had an overall better demonstrated representation of the Pareto front. With the demonstrated utility of the methodology in hand, the subsequent chapters will proceed to apply this methodology to first a simple robotic scenario (Chapter 5) and finally to a large demonstration HRT cooperative team configuration selection scenario (Chapter 6). It is anticipated that the methodology will continue to prove its utility, though the role that the different application domain will play in the level of differentiation of the results is yet to be seen.

Chapter 5

Case Study: Robotic Reconnaissance Rovers

This experiment sought to analyze the performance analysis schemes utilized by several researchers to compare and contrast different candidate systems in an overall performance analysis. Specifically, the work by Tunstel [96] and Schreckenghost [85] were analyzed in great detail for this experiment. Through this analysis, the weaknesses and drawbacks of these methods were identified. By comparing four different robotic reconnaissance rovers, it was demonstrated that the state-of-the-art is insufficient to produce a meaningful, quantitative, comprehensive performance analysis. This dissertation's performance analysis methodology was applied to this robotic reconnaissance rover case study to observe how it affected the design space and the resulting solutions.

The idea of utilizing a composite task score to aggregate performance metrics was first proposed in the HRI realm by Rodriguez [74]. This topic was discussed in detail in section 2.2.4. Rodriguez's highly cited work proposed using relative measures for each of the performance metrics to allow comparison between metrics of different scales and units, and combining each of these ratios in a linear uniform summation to create an overall performance score.

The works by Tunstel [96] and Schreckenghost [85] were analyzed in great detail for this experiment. Tunstel's paper used Rodriguez's approach to a composite task score to assess overall operational performance of the Mars Exploration Rovers (MERs) Spirit and Opportunity during a specified window of activity. Schreckenghost's work focused on reporting the results of Earth-analogue field testing of the K10 mobile robotic systems designed for robotic reconnaissance work. Each of these works will be briefly summarized in the sections to follow.

It was the stated goal of both of these works to demonstrate that performing reconnaissance tasks with robotic technology would be helpful for future space missions. Ideally, it should be possible to compare the performance of both the MERs and the K10 rovers to assess their relative performance and success at a variety of tasks. However, the metrics used in the two papers were vastly different and do not overlap. This is an example of a major short-coming of the reporting of performance metrics from the HRT realm. The data provided in the papers, however, was sufficient to allow relative comparison in this case study and to facilitate the application of this dissertation's methodology.

5.1 Summary of Tunstel's Method and Results

Tunstel [96] applied Rodriguez's [74] relative performance ratios and composite task score method to assess the operational performance of the Mars Exploration Rovers (MERs) Spirit and Opportunity. He defined five performance metrics for use in his analysis: total autonomous traverse distance, terrain-based autonomous navigation speed, approachability, instrument placement position accuracy, and instrument placement repeatability. Tunstel defined approachability as the distance traveled (d_{app}) and time ($sols$) it took to reach a specified number of targets (N), or

$$Approachability(m/sol) = \frac{1}{N}\sum_{i=1}^{N}\frac{d_{app}}{sols}. \tag{5.1}$$

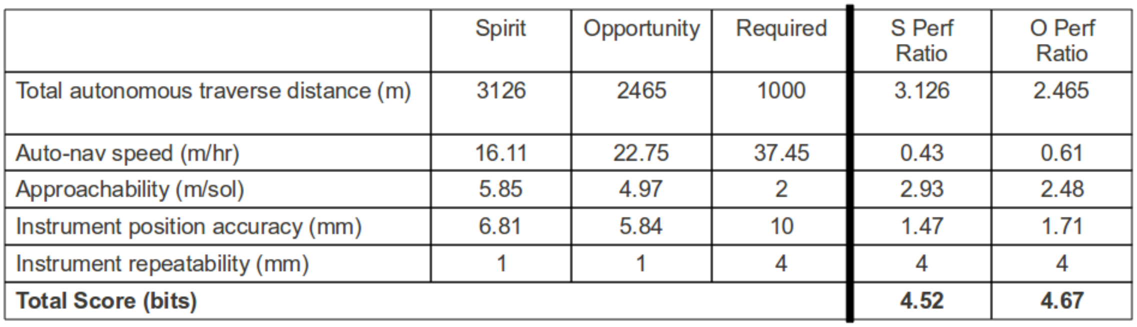
While Tunstel's first three metrics are maximization metrics (higher value is better), the last two metrics are minimization metrics (lower value is better). To accommodate for this, Tunstel changed Rodriguez's performance ratio formula such that for maximization metrics it would still be calculated as performance score/reference score, but for minimization metrics the ratio would be inverted: reference score/performance score.

Rather than using a strict linear summation of the performance ratios, Rodriguez and Tunstel both preferred to use a base-2 logarithm summation of the performance ratios represented as

$$Score(r) = \frac{1}{2}\{log_2[P^2(m_1, r)] + log_2[P^2(m_2, r)] + +log_2[P^2(m_N, r)]\} \tag{5.2}$$

The resulting value from the base-2 logarithm is in bits, allowing the score to be read as "rover A performed five times better than required".

As can be seen in equation 5.2, each performance metric in Rodriguez's formula is represented as a ratio of a system's score to the reference score in that metric. The reference system used in Tunstel's calculations were the MER's operational requirements. For the MERs, these requirements were an autonomous traverse distance of 1000 meters, an autonomous navigation speed of 37.45 meters/hour, approachability of 2 meters/sol (where a sol is a Martian day), an instrument placement

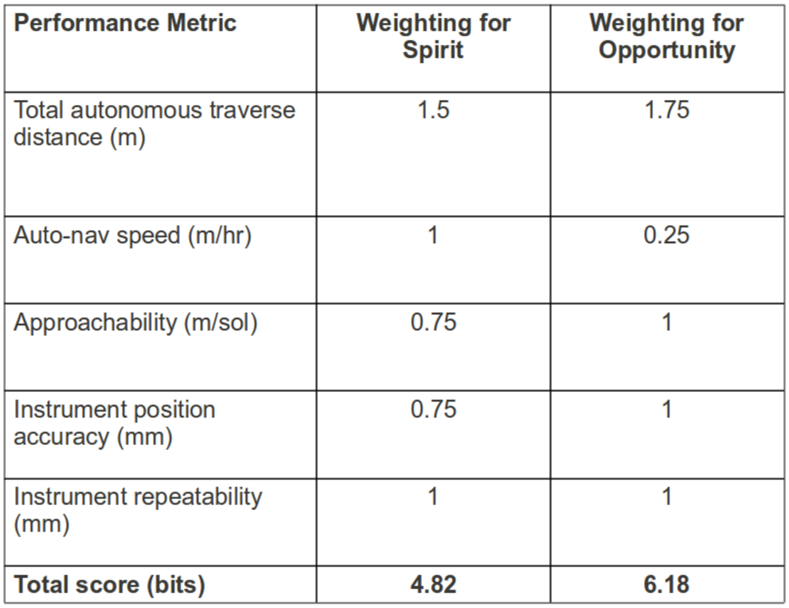


	Spirit	Opportunity	Required	S Perf Ratio	O Perf Ratio
Total autonomous traverse distance (m)	3126	2465	1000	3.126	2.465
Auto-nav speed (m/hr)	16.11	22.75	37.45	0.43	0.61
Approachability (m/sol)	5.85	4.97	2	2.93	2.48
Instrument position accuracy (mm)	6.81	5.84	10	1.47	1.71
Instrument repeatability (mm)	1	1	4	4	4
Total Score (bits)				**4.52**	**4.67**

Figure 5.1: Performance Metrics, Performance Ratios, and Overall Scores for Mars Exploration Rovers Spirit and Opportunity. Data reproduced from [96].

position accuracy of 10 mm, and an instrument placement repeatability of 4 mm [96]. The results from each of the five metrics and the final score in bits as derived from Tunstel's research is reproduced in Figure 5.1 for both Spirit and Opportunity.

Tunstel acknowledged a shortcoming of Rodriguez's method was the equal weighting of the performance metrics, which presumes equal importance of the metrics in the performance analysis. In a first attempt to reflect a difference in mission priorities between the metrics and between the MERs, Tunstel used weightings to specify the different mission characteristics of Spirit and Opportunity. These weights (shown in Figure 5.2) were used to reflect the different environments in which the rovers were placed, and how this affected the overall mission objectives and desirable performance characteristics. Tunstel did not provide justification for these scalar values. This was an arbitrary assignment of weightings. Using the weightings,



Performance Metric	Weighting for Spirit	Weighting for Opportunity
Total autonomous traverse distance (m)	1.5	1.75
Auto-nav speed (m/hr)	1	0.25
Approachability (m/sol)	0.75	1
Instrument position accuracy (mm)	0.75	1
Instrument repeatability (mm)	1	1
Total score (bits)	**4.82**	**6.18**

Figure 5.2: Weightings for each of the Performance Metrics and Weighted Scores for Mars Exploration Rovers Spirit and Opportunity. Data reproduced from [96].

Tunstel claimed that although Opportunity received only a slightly higher overall score in the unweighted case, Opportunity should have received a significantly higher overall score, as seen in Figure 5.2.

Tunstel provided additional metrics in [96] that he did not use in his performance analysis of the MERs. Tunstel did not provide justification for the selection of some metrics and not others. The five metrics described but not used in Tunstel's analysis are summarized here for consideration in section 5.3. Tunstel defined *autonomous traverse speed ratio* to be the ratio of the average autonomous navigation speed to the maximum autonomous navigation speed for a given system. In other words, this metric would create a scaled value of autonomous navigation speed that reflects the relative performance of a vehicle with reference to its own capabilities rather than to an external reference.

Navigation step time was defined to be the time required for the autonomous components to perceive the local terrain, detect hazards, and drive 35 cm along a chosen path. *Percent autonomous traverse* would provide a metric to gage how much of a mission was performed autonomously. *Mean self-localizations per sol* represents the number of position updates of the rover in each Martian day, and *mobile manipulability* is the ratio of robotic arm workspace volume to the mobile platform volume.

5.2 Summary of Schreckenghost's Method and Results

Schreckenghost [85] summarized the findings of field trials of the NASA Ames K10 rovers performed in 2008 at Black Point Lava Flow, AZ. One of the limiting factors to including robotics in space operations to date has been the need to provide custom technology for each mission. The K10 rovers were intended as a demonstration of commercial off-the-shelf technology, to provide an example of the fact that a rover not specifically designed for the mission could still be very useful in surface mapping and other reconnaissance activities. The K10 rovers are 80 kg, 4-wheel drive rovers with all wheel steering and a passive rocker suspension, capable of carrying an additional 15 kg of payload each. They are capable of driving up to human walking speeds (90 cm/s) [30].

In [85], the findings from the Black Point Lava Flow field trials were averaged for each day of operations, such that mission time was represented as a percentage.

Daily productive time was assessed to determine the percentage of each day spent performing autonomous operations (average of 34% for each day, and a total of 85% of the productive time in each day), and both scheduled and unscheduled manual operations. Daily overhead time was the percentage of each day spent out of plan, in plan but inactive, or waiting for a plan to start. Daily task time was the percentage of each day during which tasks were completed, tasks were failed, or tasks from the mission plan were never attempted. This last metric requires a better explanation. During the mission, the K10 rover completed 324 of its required 449 tasks. It aborted 115 tasks before they were started (assessed that it would be unable to complete the task). Ten tasks were attempted and then not completed.

Although it was acknowledged that environmental conditions had a significant impact on the rover's performance, these effects were not quantized in Schreckenghost's analysis. The rover was given a pass/fail designation for each task attempted with no individual analysis of how well a task was performed.

To facilitate comparison with the MERs, more performance data for the K10 rovers was sought. NASA Ames has used the K10 rovers in several other field trials. Fong [30] described a three week field trial at Haughton Crater in Canada during which the research team completed 200 hours of robotic survey operations. Roughly 10% of this time was conducted autonomously. Two different models of the K10 were used during these trials - a red one, equipped with a Lidar for imaging, and a black one equipped with ground penetrating radar for subsurface mapping.

In the Haughton Crater field trials [30], the red K10 performed 9 days of operations, traversed a total of 14 km, took 25 Lidar panoramas, and traversed at speeds of 5, 10, 20, and 40 cm/s. It was assessed that traveling at speeds of 20 cm/s and 40 cm/s were too fast for the Lidar to take quality images. The black K10 performed 10 days of operations, traversed a total of 32.2 km, and found evidence of wet clay 20 cm below the dry sand.

Schreckenghost [82] described similar field trials conducted in 2008 at Moses Lake Sand Dunes, WA. The red K10 performed 28 total hours of operations and the black K10 performed 9.5 hours. The red K10 was officially powered-up (run time) for 9.35 hours but only drove for 4.2 hours. The black K10 had a 5.67 hour run time and a 0.64 hour drive time. It was stated that the difference between run time and drive time was due to a substantial wait time attributable to both poor lighting and a bad communication link. The red K10 had been planned to traverse 2235 meters but actually covered 2375 meters. The black K10 had been planned to traverse 776 meters but drove a total of 838 meters.

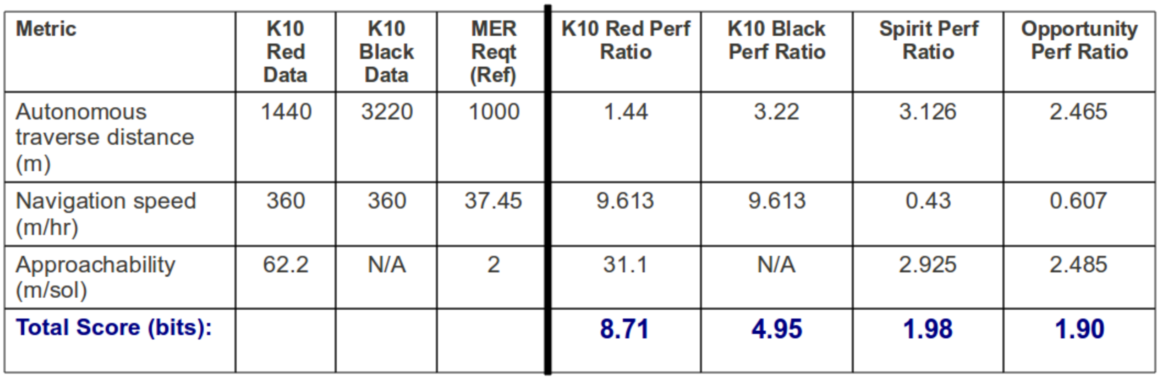
5.3 Using Tunstel's Method to Compare the Four Rovers

Both the K10 rovers and the MERs were tasked with robotic reconnaissance work and site survey operations. It would be illustrative to have a simple means of comparing the performance of the rovers. There was enough information from the combined K10 field trial data to enable application of Tunstel's performance metrics to the K10 rovers. For example, Tunstel had defined *approachability* as the number of days an approach took divided by the number of sols, summed over each of the approaches, divided by the total number of targets reached during that interval.

For the red K10, its 14 km traverse at Haughton Crater was performed in 9 days and facilitated 25 Lidar panoramas (or 25 targets being reached). According to the formula, approachability = (1/25)*(14km/9days) = 62.2 m/day. The black K10 was not equipped with an instrument that was conducive to using the approachability performance metric.

Tunstel's metrics and analysis system were applied to the K10 rovers to enable comparison between them and the MERs. Neither of the K10s had a robotic arm that was conducive to using the metrics for instrument placement, so the performance metrics that depended on instrument placement were neglected from this first analysis run. As specified in [30], Lidar imaging was only feasible at navigation speeds of less than 20 cm/s. An average navigation speed was assumed for the K10s to be 10 cm/s, or 360 m/hr. Fong had also specified that 10% of the traverse was performed autonomously. For the purposes of this analysis, therefore, the K10 red rover traveled 1400 m autonomously, and the K10 black traveled 3220 m autonomously.

Figure 5.3 shows the resulting performance ratios for each of the four rovers under consideration. With the three performance metrics applied to them, the composite task scores yielded that the K10 red rover performed 8.7 times better than the required performance for the MERs, and the K10 black performed 4.9 times better. According to these three metrics, all four rovers met their performance requirements. Spirit and Opportunity performed nearly two times better than required. It would be misleading, however, to view these numeric results as conclusive. It will be demonstrated that the composite task scores are highly sensitive to the metrics



Metric	K10 Red Data	K10 Black Data	MER Reqt (Ref)	K10 Red Perf Ratio	K10 Black Perf Ratio	Spirit Perf Ratio	Opportunity Perf Ratio
Autonomous traverse distance (m)	1440	3220	1000	1.44	3.22	3.126	2.465
Navigation speed (m/hr)	360	360	37.45	9.613	9.613	0.43	0.607
Approachability (m/sol)	62.2	N/A	2	31.1	N/A	2.925	2.485
Total Score (bits):				**8.71**	**4.95**	**1.98**	**1.90**

Figure 5.3: Robotic Reconnaissance Rover Performance Comparison. Using Tunstel's Performance Metrics and Analysis Method to Compare the Operational Performance of the K10 rovers and the MERs.

used in their computation.

It is illustrative to consider a wider range of performance metrics in the composite task score analysis. Three additional performance metrics were added for this analysis. All three of these metrics were introduced and discussed in Tunstel's paper [96], but he did not use them in his quantitative analysis. The first additional performance metric used for this next phase of analysis was *longest single sol traverse*. This metric is self-explanatory.

The data available for the K10 rovers did not facilitate use of Tunstel's instrument position accuracy and instrument repeatability performance metrics, but Tunstel had described an additional metric to assess how often an instrument was used (with the assumption that a higher instrument use rate would be beneficial).

Instrument placement rate was defined to be the number of instrument placements divided by the total number of sols it took to complete them.

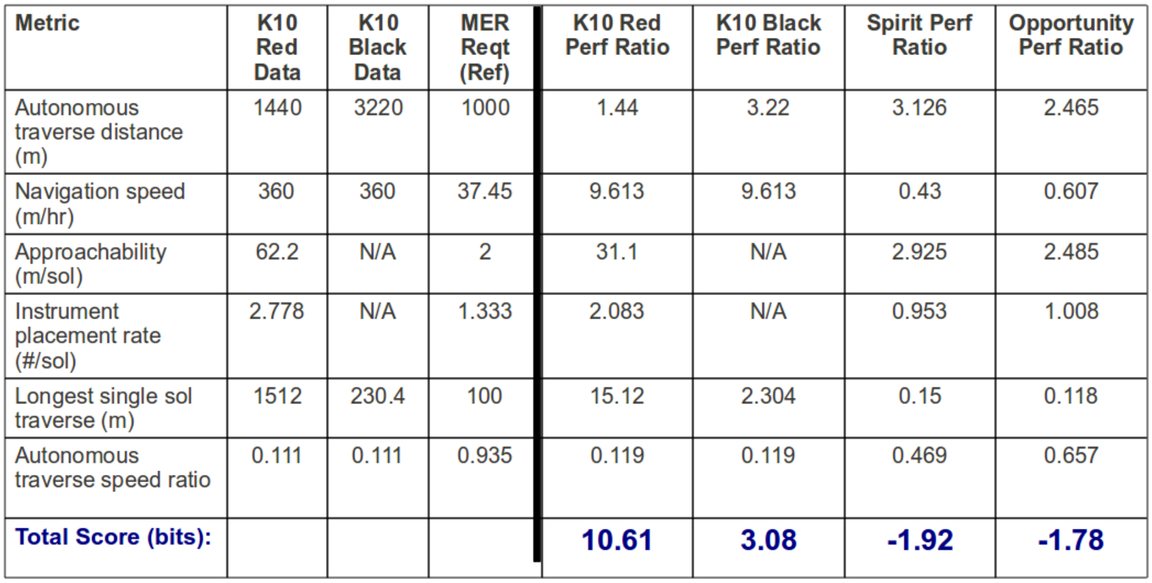
This was the second performance metric added to this analysis. Performance values for all four of the rovers needed to be calculated for this metric. K10 red performed 25 instrument placements in 9 days, resulting in an instrument placement rate of 2.778/sol. K10 black did not have an instrument conducive to this analysis. Spirit completed 1200 instrument placements in 944 sols while Opportunity completed the 1200 instrument placements in 893 sols, resulting in instrument placement rates of 1.27/sol and 1.34/sol, respectively. For the metrics that did not apply to the K10 black rover, zeros were used for its metric values for the following calculations. The MER requirement for instrument placement rate was derived from the MER requirements for 1200 instrument placements in 900 days [96], resulting in a performance requirement of 1.33/sol.

A third performance metric included in this analysis was *autonomous traverse speed ratio*, defined as the ratio of the average to the maximum autonomous traverse rate. This metric relates the performance of each rover against its own capabilities. As stated in [30], the K10 rovers were capable of 90 cm/s maximum traverse rate, or 3240 m/hr. Using an average autonomous traverse rate of 10 cm/s or 360 m/hr, this resulted in an autonomous traverse speed ratio of 0.111 for both K10 rovers. According to Tunstel, Spirit's and Opportunity's average autonomous traverse rates were 15.06 m/hr and 22.09 m/hr, respectively, and their maximum autonomous traverse rates were 34.35 m/hr and 36 m/hr, respectively. This yielded an autonomous traverse speed ratio for Spirit and Opportunity of 0.438 and 0.614, respectively. The

MER requirement for an average autonomous traverse rate of 35 m/hr and a maximum autonomous traverse rate of 37.45 m/hr resulted in a required autonomous traverse speed ratio of 0.935.

The other metrics suggested by Tunstel were not used in this analysis because there was insufficient data to well-represent the four candidate rovers. *Navigation step time* and *mean self-localizations per sol* required information about processing time and position updates of the autonomous software on-board each rover. Insufficient data was available to accurately describe the *mobile manipulability* for the MERs and the K10s. *Percent autonomous traverse* was not included in this analysis because it would be comparing apples to oranges - in their field trials, the mission plans for the K10 rovers did not use them as entirely autonomous rovers. They had periods of autonomous traverse, but portions of their mission schedule were intended to be teleoperated. Although all portions of the MER traversals were conducted autonomously, there were large intervals of wait time while a new set of commands were being uploaded. Comparing the percentage of the K10 missions conducted autonomously to the MERs' would misrepresent their intended missions.

The results from the expanded performance metric analysis of the four candidate rovers were tabulated in Figure 5.4. Examining the overall composite task scores for the four rovers, the K10 rovers continued to out-perform the MERs. According to this analysis, the red K10 rover proved 10.6 times more capable than required by the MER requirements. Surprisingly, the black K10 rover proved 3 times more capable than required even though it failed to score in two of the performance metrics. This suggests a flaw in the analysis method - if a rover was incapable of

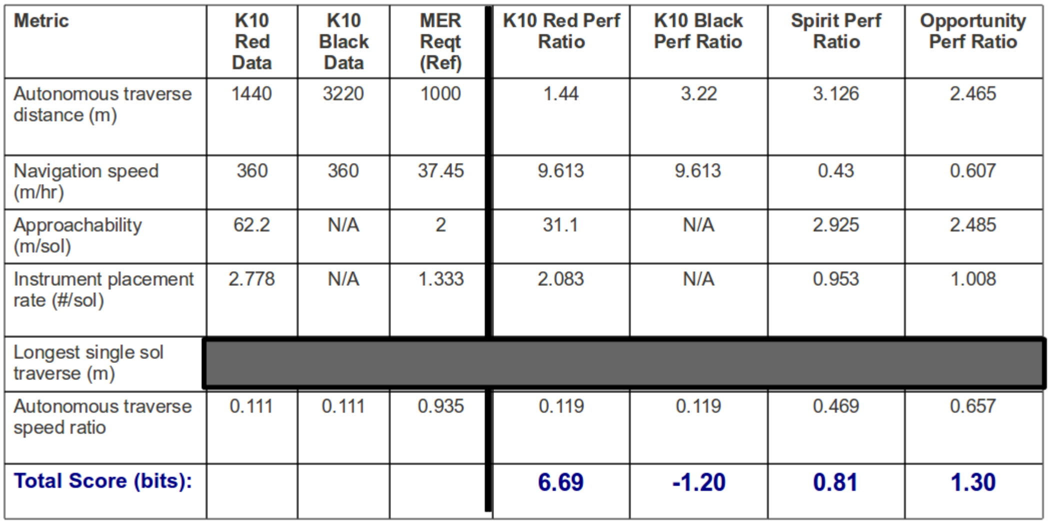


Metric	K10 Red Data	K10 Black Data	MER Reqt (Ref)	K10 Red Perf Ratio	K10 Black Perf Ratio	Spirit Perf Ratio	Opportunity Perf Ratio
Autonomous traverse distance (m)	1440	3220	1000	1.44	3.22	3.126	2.465
Navigation speed (m/hr)	360	360	37.45	9.613	9.613	0.43	0.607
Approachability (m/sol)	62.2	N/A	2	31.1	N/A	2.925	2.485
Instrument placement rate (#/sol)	2.778	N/A	1.333	2.083	N/A	0.953	1.008
Longest single sol traverse (m)	1512	230.4	100	15.12	2.304	0.15	0.118
Autonomous traverse speed ratio	0.111	0.111	0.935	0.119	0.119	0.469	0.657
Total Score (bits):				**10.61**	**3.08**	**-1.92**	**-1.78**

Figure 5.4: Robotic Reconnaissance Rover Performance Comparison. Tunstel's Expanded Performance Metrics Applied to the K10 Rovers and the MERs.

meeting a performance requirement, a penalty should have been applied. It should not have been mathematically possible for this rover to achieve an overall score that was better than required.

It is interesting to note that neither of the MERs met their overall required performance composite score as calculated in Figure 5.4. Although both MERs performed well in total autonomous traverse distance and approachability, they did not meet the required performance benchmarks for navigation speed, instrument placement rate, longest single sol traverse, and autonomous traverse speed ratio. The result of not meeting these metrics was evident in their composite task score values. The negative score represents the fact that, according to this set of metrics, Spirit and Opportunity performed nearly two times worse than required.



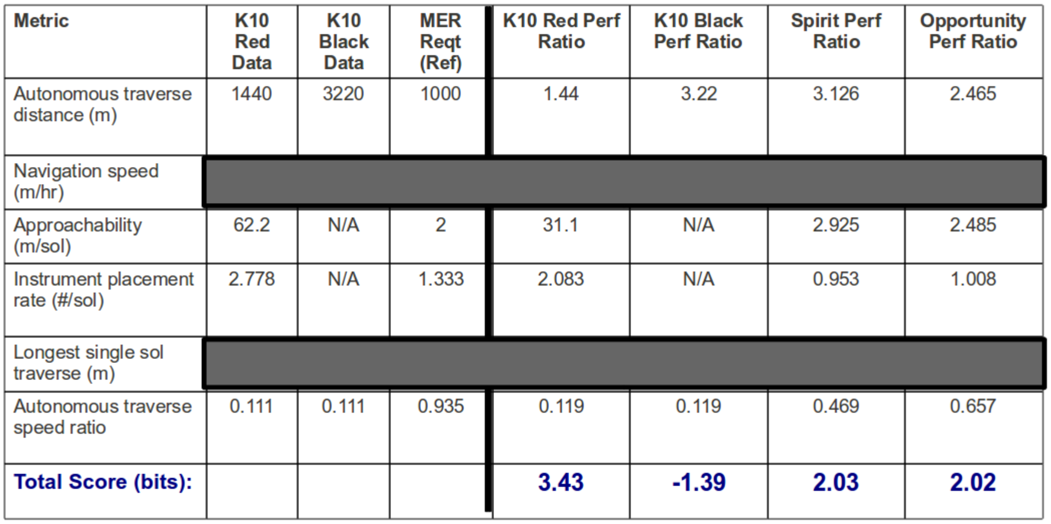
Metric	K10 Red Data	K10 Black Data	MER Reqt (Ref)	K10 Red Perf Ratio	K10 Black Perf Ratio	Spirit Perf Ratio	Opportunity Perf Ratio
Autonomous traverse distance (m)	1440	3220	1000	1.44	3.22	3.126	2.465
Navigation speed (m/hr)	360	360	37.45	9.613	9.613	0.43	0.607
Approachability (m/sol)	62.2	N/A	2	31.1	N/A	2.925	2.485
Instrument placement rate (#/sol)	2.778	N/A	1.333	2.083	N/A	0.953	1.008
Longest single sol traverse (m)							
Autonomous traverse speed ratio	0.111	0.111	0.935	0.119	0.119	0.469	0.657
Total Score (bits):				**6.69**	**-1.20**	**0.81**	**1.30**

Figure 5.5: Robotic Reconnaissance Rover Performance Comparison. Tunstel's Performance Metrics Applied to the K10 Rovers and the MERs. Redundant Distance Metric Removed From Composite Task Score.

It could be argued, however, that these six metrics contain redundant information and refer to correlated data. Two of the metrics refer to distance traveled, and two of the metrics refer to autonomous traverse speed. To analyze the effect that these correlations may have, a traverse metric (longest single sol traverse) was removed from consideration and the numbers recalculated. These results were tabulated in Figure 5.5. According to this set of analysis, the K10 red rover continued to outperform all other rovers. Spirit exceeded expectations by 80% and Opportunity by 130%. The K10 black rover, however, did not meet the MER requirements. Its overall composite task score demonstrates that it performed 120% times worse than required.

For demonstration purposes, a further reduction in performance metrics removed one of the speed metrics (navigation speed). The results from this analysis are tabulated in Figure 5.6. As can be seen, Spirit and Opportunity appeared to have had better performance with this set of metrics. Both MERs performed twice as well as had been required. K10 red performed 3.5 times better than had been required, and K10 black performed 1.3 times worse than required.

All four rovers had different composite task scores in each of the four sets of analysis that used Tunstel's methodology and performance metrics. Not only were the quantitative values themselves variable, but the relations between the rovers and the requirements were variable. While the K10 red rover outperformed all of the other rovers in all of the cases, the ranking of the other three rovers was dependent on the set of performance metrics used in each analysis. Even the ability of the rovers to meet their design requirements varied between each of the sets of analysis.



Metric	K10 Red Data	K10 Black Data	MER Reqt (Ref)	K10 Red Perf Ratio	K10 Black Perf Ratio	Spirit Perf Ratio	Opportunity Perf Ratio
Autonomous traverse distance (m)	1440	3220	1000	1.44	3.22	3.126	2.465
Navigation speed (m/hr)							
Approachability (m/sol)	62.2	N/A	2	31.1	N/A	2.925	2.485
Instrument placement rate (#/sol)	2.778	N/A	1.333	2.083	N/A	0.953	1.008
Longest single sol traverse (m)							
Autonomous traverse speed ratio	0.111	0.111	0.935	0.119	0.119	0.469	0.657
Total Score (bits):				**3.43**	**-1.39**	**2.03**	**2.02**

Figure 5.6: Robotic Reconnaissance Rover Performance Comparison. Tunstel's Performance Metrics Applied to the K10 rovers and the MERs. Redundant Distance and Speed Metrics Removed From Composite Task Score.

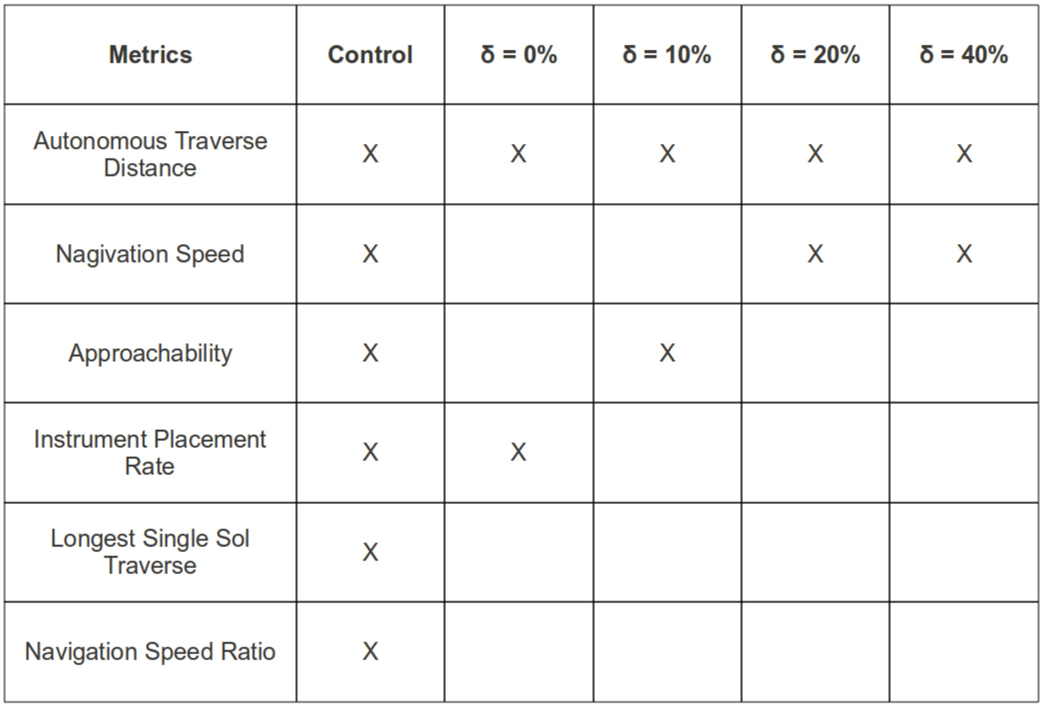
It is clear from this case study that Rodriguez's [74] composite task score is highly sensitive to the performance metrics used. Adding and removing metrics drastically changed the composite task score values. This sensitivity, however, is not warranted. It results in a highly subjective ranking of the rovers. The performance metric set used in any given analysis could be altered to produce a preconceived notion of the intended results.

It is desirable to find a method to gage the relative performance of the four rovers that is less dependent on the performance metrics selected. Toward this goal, this dissertation's methodology was applied to the robotic reconnaissance rover case study.

5.4 Application of this Dissertation's Methodology to the Robotic Reconnaissance Rover Comparison

Application of this dissertation's methodology to this problem had the potential to provide the rigorous performance analysis required to objectively and conclusively compare the rovers. The methodology uses the δ-MOSS objective reduction algorithm at its core to determine the importance of each of the objectives (or performance metrics) depending on the amount of information each provides. A pairwise comparison of the performance metric values would yield the underlying dominance structure for the problem. Using this rigorous structure, selection of performance metrics (and identification of redundancy) could be done in a non-arbitrary manner.

The δ-MOSS algorithm was applied to the four candidate rovers across the six



Metrics	Control	δ = 0%	δ = 10%	δ = 20%	δ = 40%
Autonomous Traverse Distance	X	X	X	X	X
Nagivation Speed	X			X	X
Approachability	X		X		
Instrument Placement Rate	X	X			
Longest Single Sol Traverse	X				
Navigation Speed Ratio	X				

Figure 5.7: Robotic Reconnaissance Rover Performance Comparison. Effect of δ-MOSS Objective Reduction Algorithm and Tolerated δ-Error on Performance Metric Subset.

performance metrics described in the previous section. It was determined that while performance ratios were illustrative for Rodriguez's methodology, they would obfuscate rather than aid in the clarity of the algorithm. Instead, the full performance score values were used in this analysis, scaled such that the difference between the largest and smallest value would yield a δ-error value of 1. The resulting δ-non-conflicting performance metric subsets are tabulated in Figure 5.7.

It is intriguing to note that even with zero δ-error, a large reduction in performance metrics resulted. The six metrics were reduced to two: autonomous traverse distance and instrument placement rate. These two metrics contain conflicting information that fully represent the design space. According to the first, Spirit outperformed K10 black, which in turn outperformed Opportunity, with K10 red coming in last place. According to the instrument placement rate metric, K10 red performed best, followed by Opportunity, Spirit, and K10 black. All of the data contained in the other four performance metrics contains these pairwise performance relations and no new data. These four metrics contain only redundant information, and could be removed from consideration without affecting the underlying dominance structure of the problem.

The four values of δ-error applied in the knapsack problem application (and by Brockhoff in [12]) were used in this analysis, supplemented by an analysis of which exact values of the δ-error caused an alteration to subset composition. The $\delta = 0\%$ subset was valid for only that error tolerance. When a 1% δ-error was included in the underlying dominance structure, instrument placement rate was replaced by approachability in the reduced performance metric set. When a 20% δ-error was included, approachability was replaced by autonomous navigation speed in the reduced set. This subset was unchanged as the δ-error value was increased.

It was not until a δ-error of 47% was applied that the reduced set was altered again. Navigation speed was removed from consideration, such that the reduced metric set contained only autonomous traverse distance. In other words, with a 47% error in the underlying dominance structure, the rover comparison problem devolved into a single performance metric case. Enough problem data had been removed that there was no longer conflict between the performance metrics.

Application of the δ-MOSS objective reduction algorithm provided this case

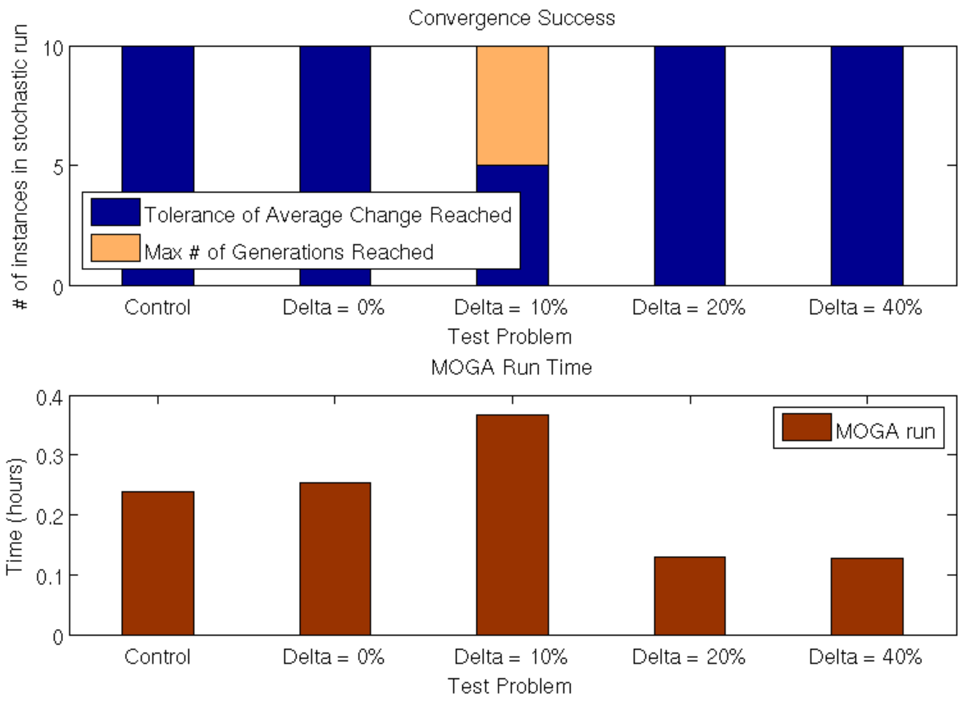


Figure 5.8: Robotic Reconnaissance Rover Performance Comparison. Convergence Success of MOGA Runs and MOGA Run Time for Each δ Value.

study with its desired result - an objective, non-arbitrary, rigorous reasoning to select some performance metrics and neglect others. This algorithm removed the arbitrary selection that had been the basis of Rodriguez's performance analysis and replaced it with a concrete method to ensure that the underlying problem definition remained constant.

To fully demonstrate the utility of this dissertation's methodology, the rest of it was applied to the robotic reconnaissance rover comparison problem. Using the reduced objective sets obtained from the δ-MOSS algorithm (the five test problem depicted in Figure 5.7) were run through the MOGA for 10 stochastic runs of 400 generations each to generate a Pareto solution set. The results are collated in the figures below.

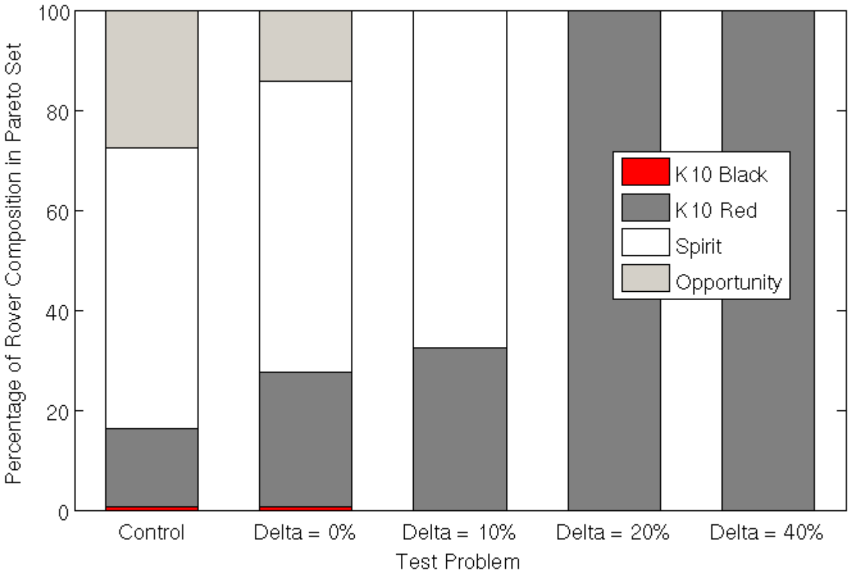


Figure 5.9: Robotic Reconnaissance Rover Performance Comparison. Final Pareto Set Composition for Each Test Problem.

Figure 5.8 shows that the MOGA had 100% convergence success in four of the five test problems. In the 10% problem instance, however, the MOGA had only a 50% success at converging to a Pareto front. Corresponding to the lengthened search process of this test case, the run time duration for the MOGA across these 10 stochastic cases was significantly longer than any of the other four test cases. The two metrics in the reduced set for this test case presented a strongly conflicting trade-off between the objective values such that the average tolerance change in the Pareto front never dropped to the convergence level.

Figure 5.9 shows the final Pareto solution results for the robotic reconnaissance rover comparison. For each of the five test cases, the bar plot demonstrates the composition of the Pareto front. In other words, the Figure demonstrates the percentage of the Pareto set for which each of the four candidate rovers represented the best overall solution. It should be remembered that due to the conflict in the performance metrics, a single candidate solution was not the anticipated result. Instead, a Pareto front of equally-scoring candidate solutions was expected.

For the first three test problems, Spirit was selected as the best candidate rover 72%, 56%, and 68% of the time, respectively. In the control case (with all 6 performance metrics), Opportunity ranked second-best (28% of the Pareto set) with the K10 Red rover following in third (16%). With the $\delta = 0\%$ test case pruning down to two performance metrics, the K10 red rover and Opportunity switched their overall rank (28% and 14%, respectively).

The K10 black rover represented a small fraction of the Pareto population (less than 1% of the total population) for both the control and $\delta = 0\%$ test cases, and was not present in the solutions thereafter. Recall that the K10 black rover did not have an instrument-arm that enabled approachability and other instrument-driven metric analysis. For these types of performance metrics, a zero value had been applied to the K10 Black rover to distinguish that it was incapable of performing some of the tasks. It is reassuring to see its lack of representation in the final Pareto sets of each of the test cases because, as an ill-equipped rover, it did not represent a good solution option. That it is present in two of the five Pareto sets, however, demonstrates that the tasks the K10 black rover is capable of performing are performed very well.

With the reduced objective function set of the $\delta = 10\%$ test case containing metrics for autonomous traverse distance and approachability, the MOGA narrowed the design space down to an option of the two best performing rovers: Spirit (68%)

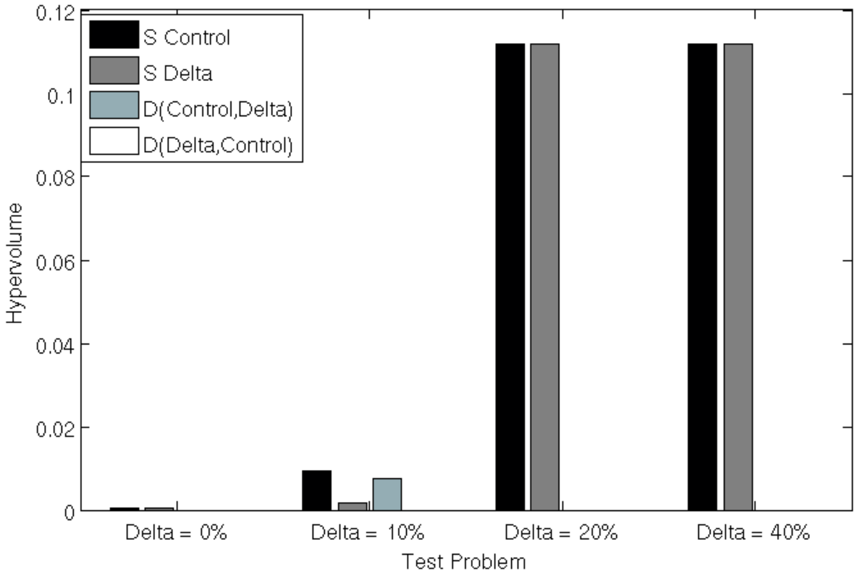


Figure 5.10: Robotic Reconnaissance Rover Performance Comparison. Hypervolume Indicator Pareto Set Quality Metric For Each Test Case.

and the K10 red rover (32%). When the δ-error was increased to 20%, however, the K10 red rover became the best overall performing rover, representing 100% of the Pareto population.

A final examination of the Pareto solution sets was conducted by using the hypervolume indicator quality metric (for a detailed discussion of the hypervolume metric, refer to section 3.2.4). Figure 5.10 contains the calculations from this quality metric. There are several important details to note in this figure. Recall that the S quality metric measures the absolute volume coverage of a Pareto set as an indication of the diversity and spread of solutions. Comparing this metric for each of the δ test cases against its corresponding control, it can be seen that in the $\delta = 0\%$, 20%, and 40% test cases, the S metric calculations for the test cases were equally as good as the control cases. In these same test cases, the D metric (which provides a relative coverage between two Pareto sets) calculated to exactly zero for both *D(control,delta)* and *D(delta,control)*.

This last finding means that neither the control Pareto fronts nor the δ Pareto fronts dominated each other at any point. With identical S metric calculations and zero values for the D quality metrics, there is sufficient information to state that the δ Pareto sets for the rover reconnaissance problem were neither better nor worse in any aspect than their control Pareto sets.

It was only in the $\delta = 10\%$ test case that the control Pareto set yielded slightly better solutions than the δ test case. As seen from the *D(control,delta)* value for this test case, the control Pareto set dominated just under 1% of the δ Pareto front. In other words, by using the δ Pareto set for the 10%-error test case instead of the control Pareto set, a 1% decrease in the quality of the solution set resulted. This decrease in quality, however, afforded a decrease in the number of performance metrics from 6 down to 2 metrics. Additionally, this slight decrease in solution quality allowed the reduction from four candidate rovers down to two candidate rovers: the K10 red and Spirit.

Not only did application of this dissertation's methodology quantitatively and objectively reduce the number of performance metrics required for the overall performance analysis, but including varying levels of error in the underlying dominance structure allowed the visualization of the best candidate solutions for the overall design problem to be considered.

5.5 Summary

This case study sought to rigorously compare the operational performance of the Mars Exploration Rovers (MERs) Spirit and Opportunity against the operational performance of two NASA Ames Earth-analogue robotic reconnaissance rovers, the K10 red and black. It was demonstrated that Rodriguez's [74] composite task score is highly dependent on the performance metrics selected and was insufficient for this rigorous heterogeneous performance analysis. To remedy this, the methodology proposed in this dissertation was utilized to reduce the performance metrics to a subset that maintained the underlying dominance structure of the problem. In other words, a reduced set of objectives resulted in no change to the problem details and the final solution set.

With zero error in the dominance structure, the six performance metrics used in this simplified analysis were reduced to two conflicting performance metrics. Four of the six performance metrics contained only redundant information and could be removed from analysis without affecting the problem and its solutions. If a high enough δ-error were tolerable in the problem, the dominance structure of the robotic reconnaissance problem was demonstrated to be reducible to a single performance metric.

The success of this dissertation's methodology to objectively and quantitatively reduce the complexity of a multi-objective optimization problem has been demonstrated on both the knapsack application, and on the comparison of rovers for a robotic reconnaissance mission. The δ-MOSS algorithm has proved itself to be a powerful tool to reduce problem complexity by focusing the analysis on relative performance relations rather than on relative numbers. This algorithm, synthesized with the other components described in Chapter 3, have created a generic, rigorous, objective, quantitative methodology for the HRT domain to both select the performance metrics to be used in an overall team performance analysis and to reduce the complexity of a mission designer's final decision space. This type of methodology is novel for the domain and a valuable contribution to further the use of human and robot teams. It is anticipated that this dissertation's proposed methodology will prove to be a valuable tool in human-robot team configuration selection problems, the intended application domain of this methodology.

Chapter 6

Large HRT Application Demonstration

The heterogeneous HRT configuration selection problem was the motivating application domain for the methodology developed in this research. In essence, the question is how to compare the operational task performance of heterogeneous teams across many different performance metrics, without necessitating preference information, and without dictating an equal weighting (or other arbitrary method of weighting) the performance metrics. The overall goal is to input a mission or set of tasks, several dozen possible team combinations (each with pre-defined task capabilities), and output the overall team performance in a manner that facilitates a mission designer's final configuration selection.

In this final experiment, a large-scale HRT configuration selection problem was developed. This dissertation methodology was applied to the design space and greatly reduced the complexity. The Pareto sets that were generated across the five test problems were compared. A final set of analysis was conducted to determine the repeatability of the solutions.

6.1 Initial Setup: Hubble Space Telescope Servicing Mission 3A

For this experiment, the mission profiles from the Hubble Space Telescope Servicing Mission 3A were selected to form the background of this performance analysis. HST Servicing Mission 3A (HST SM-3A) provides a useful platform to analyze role definition of each of the mission agents, influences of task allocation schemes, the resulting cooperative schedules, and the overall performance of the combined human and robot servicing team.

The flight plans from past Hubble Space Telescope servicing missions (HST SMs) (as represented in the HST SM-3A EVA Checklists [100]) provide a detailed data set from which to examine the effect of various activities and crew performances on space operations. HST was designed with access panels to allow repair of the components of the telescope. Some of the servicing tasks required a fine level of dexterity, including manipulation of tethers and electrical connectors.

Two astronauts were involved in each EVA excursion - one positioned on the end of the Space Shuttle's robotic arm, the remote manipulator system (RMS), and one free floater who either tethered to various parts of HST during task performance or used the portable foot restraint (PFR) to anchor to a worksite.

Both of the humans were obligated during all of the tasks in varying capacities. Generally, the astronaut positioned by the RMS (called EVA2) performed hardware maneuvering and use of the power tools, while the other astronaut (EVA1) disconnected securing straps and connectors from hardware and provided supporting aid when needed. Replacement units were transferred from one astronaut to another. Both astronauts were used to carefully position instruments like the fine guidance sensor (FGS) along guide rails for insertion and installation into HST. Only a few of the subtasks involved with HST SM-3A required manipulation of electrical connectors or harnesses.

Four days of EVA were planned for this mission. The 6-hour nominal EVA day (including daily setup and closeout of HST worksites) was the primary constraint on the number of tasks that could be performed during the servicing mission. This constraint was a NASA rule based on environmental exposure and consumables. In addition, there were sequential constraints between some of the subtasks (e.g., a team member cannot open a door before its securing bolts have been unscrewed).

Most of the tasks can be broken down into primitives that require single-degree-of-freedom motion and utilization of a single tool [68]. Although repetitive single-tool primitives require a significant amount of the human crew's time to perform, the skills needed to perform these types of tasks are well within the capabilities demonstrated by Robonaut or the Mars rovers. It would be anticipated that if robots were used as cooperative team members tasked with performing this type of primitive that the human crew could be freed to work on other, more specialized tasks. This suggests the potential for improvements in crew utilization by including robots in operations as cooperative team members.

The HST SM-3A mission is used in this research as an example mission for which a wealth of primary source data is available. Although the mission took place in the zero-gravity space environment, a terrestrial servicing mission could require similar tasks. The mobility differences between zero-gravity and other gravity spectrum would be anticipated to affect the time intervals required to complete each subtask. To avoid adding additional sources of error and uncertainty, the subtask performance completion times from the EVA checklist were used in this research.

6.1.1 Task Primitive Data Preparation

The initial steps to create a human and robot team's schedule follow the process of nominal schedule development. The target set of tasks to be accomplished must be identified and decomposed into task primitives. Following the procedure for the three-tier hierarchical task analysis from the NASA mission timelines [100], each task (eg. instrument replacement) is decomposed into subtasks (eg. retrieval of an instrument), which are further segmented into the primitives (eg. unlatching a door) that define the specific steps needed to complete a given task.

Pilotte [68] analyzed each of the task primitives from HST SM-3B, the follow-on mission to HST SM-3A, with the goal of assessing what types of robotic end effectors would be needed to complete the mission. In her analysis, Pilotte categorized each of the task primitives according to which end effector would be required to perform it. Following her methodology example, each task from HST SM-3A was analyzed for this research at the primitive level to determine which of nine different task primitive categories the item belonged (multiple categories was an option): visual inspection, mixed-initiative verbal communication, translation between worksites, carry or transport of instruments and hardware between locations, use of a handrail-gripper tool, use of a pinch-grasp type of tool, a force-push (requires the ability to counterbalance the applied force), a force-pull, and use of an additional specialized tool.

A database was created that contained the human subtask performance times from the HST SM-3A data, with the addition of a numeric representation of the types of primitives that were required for the completion of each subtask. Each of these subtasks were then assessed to determine whether they were constrained to follow other subtasks. If so, precedent constraints were added to the database for each subtask.

The task scenarios used in this research involved the robotic agents participating in the motor activities of the servicing mission but not in the cognitive or sensory parts of the mission. The NASA HST SM-3A mission was highly-scripted. The tasks that were analyzed as part of this mission were primarily reliant on motor skills with a few subtasks involving visual inspection of the worksite. For this research, this meant that the robots in each of the scenarios were able to contribute significantly to the entire mission task list. If the task list had included the perceptual and cognitive tasks performed by both the mission planners and enacted by the EVA crew (and a teleoperator), the ratio of robot subtask involvement would have been reduced, but the contribution to the overall task performance and team efficiency would remain unchanged.

The actual rate of task performance for a specified robotic agent will vary for different categories of tasks, and between iterations of the same task. As with humans, a robot might not perform a task at the exact same rate on two different attempts. There will be competing priorities in task performance including power and energy resource utilization, measurement errors and overshoot correction, differences in environmental factors including lighting, situational awareness, and maintaining safe operating conditions in a crowded workspace. All of these factors will affect the robot's actual task performance speed and will vary between tasks.

The human subtask completion time data used in this analysis was identical to that originally anticipated from the HST SM-3A mission. However, some variability was added to the robot time data to observe the affect on the overall team schedules and performance if the robots performed tasks at the same speed as their human counterparts, at half the speed of the humans, and if their performance required three times as much time as the humans.

The robot time data values used in this analysis provide an initial guide to enable scheduling analysis for a cooperative human and robotic team. These values could be updated to reflect a specific robotic system design if desired by future users.

6.1.2 Team Characterization and Scheduling

Eighty-two unique teams were generated for this research. Each team had at most four agents, two human and two robotic. Each of the agents in these teams were defined based on which of the nine task primitive types they were capable of performing, and the rate of robot task performance (equal-to, two-times, or three-times as slow as a human). The teams were specified with a matrix for each, with the rows corresponding to the four possible agents, and each column representing a task primitive type. A value of one in the matrix meant that the specified agent could perform the task primitive type. A value of zero meant that the agent did not have the skill set or capacity to perform the task primitive type.

The humans and robots on each of these teams were assumed to be collocated, sharing the same work environment. Team pairings that required the humans and robots to function in the same workspace were assumed to have the necessary safety systems implemented. Amongst the teams, different kinds of task specialization were selected. In some of the teams, the robotic agents were only capable of carrying hardware and translating between locations. This type of task allocation placed the robots in the astronaut-assistant role in the mission design. Alternatively, in some of the teams at least one of the robotic agents was fully capable of all of the task primitive types. This type of robot, similar in capabilities to a mobile version of NASA's Robonaut, contributed to completing mission tasks on a level that a third human would have done.

These team task capability matrices were input into a simple task allocation function written for this research. Preference was given to the robot agents when allocating tasks. In other words, if a robot was capable of completing the subtask (could complete each of the required primitive types for the subtask), that subtask was allocated to the robot. Subtasks that required two agents to complete it in SM-3A also required two agents to complete it in this research. The task allocation scheme took this into consideration, and allocated two agents to a specified subtask when needed.

As the task allocation was completed *a priori* and not altered thereafter, no preemption was allowed within the schedules. Once an agent was assigned a task, that agent had to complete that task before moving on to another. Adding preemption into a real-time mission, or allowing agents to simultaneously work on two different tasks at the same time would facilitate more elaborate and interlaced schedules. Neither of these paths were pursued in this research effort. A simple, straightforward implementation of team schedules would lend itself to better demonstrating the impact of this dissertation's methodology as described in Chapter 3.

With similar reasoning, a more advanced task allocation scheme and scheduling paradigm could be utilized in future work. There has been substantial research into both of these areas in the literature. For example, the lunar mission planning software package (HURON (Human-Robot Task Network Optimization) [26]) developed at JPL could be used to facilitate task allocation, planning, and scheduling for combined human and robotic activities. It provides the architecture to develop optimal task allocation and scheduling for a scripted multi-agent lunar mission.

The task allocation and scheduling methods used in this research do not seek optimality. Instead, they are focused on simplicity of implementation. The methods used in this research were deemed sufficient for a methodology demonstration.

Four parallel and interrelated timelines were generated by the task allocation function, one for each of the four possible agents on a team. These schedules sought to fill each agents' free time, and complete their allocated tasks as quickly as possible. Subtasks were shuffled in each agent's schedule to minimize agent downtime if the subtasks were not sequentially constrained. Precedent constraints were observed during the initial scheduling process - a subsequent subtask was constrained to begin only after its preceding subtask had been completed, regardless of which agents performed each subtask.

It is intuitive that adding a third and fourth agent to the servicing team would create schedule benefits by reducing the workload of the original two human crew through parallel execution of tasks. The level of contribution of these additional agents depended on the proportion of mission tasks that the agents were equipped to perform, the interdependence of this subset of tasks to those performed by the other crew members, and the rate of task performance and completion of the subset.

The scheduling tool rearranged subtasks within the schedule to minimize human involvement time in each mission day (freeing the astronauts to work on other tasks) and minimize the time one agent has to wait for another while maintaining the precedence constraints between subtasks.

As had been identified in [89], the primary constraint on the cooperative human and robot schedules was the nominal 6-hour EVA time window. All nominal NASA operations that involve the human crew outside of the vehicle must occur within this 6-hour window. As implemented in this research, the length of the human involvement was limited to 6 hours a day. If a team schedule required human involvement for longer than this time interval, a penalty was applied to the objective function values (please refer to the discussion of algorithms in section 3.2).

6.1.3 Performance Metrics

The performance metrics selected for this analysis were intended to measure the overall team performance of each of the heterogeneous teams, across multiple domains of interaction. To achieve this goal, several different types of performance metrics were selected. A total of fifteen performance metrics were utilized in this analysis. Each performance metric will be discussed in this section.

Limiting the analysis to fifteen performance metrics was done for several reasons. As seen in the first set of experiments with the knapsack problem in Chapter 4, all of the algorithms ran smoothly and the MOGA performed reasonably well with 15 objectives. When the number of objectives were increased to 25, however, the MOGA struggled to find an optimal Pareto front. This had been identified as a weakness of Matlab's MOGA itself. A different MOGA could be used in further research that was specifically equipped to deal with the many-objective nature of a larger HRT configuration selection problem. However, for this analysis and for the purpose of demonstrating how a large HRT configuration selection problem could be framed, using fifteen objectives was deemed sufficient.

The metrics were altered such that each sought to minimize its respective objective function. For example, to maximize the ratio of the amount of time spent actively contributing to the mission to the total mission involvement time, the objective function used would seek to minimize the amount of time spent inactive to the total mission involvement time.

A combination of task load model and expected value model was used to frame the large HRT configuration selection problem. As detailed in section 2.2.5, task load models facilitate workload considerations and inter-agent comparisons. Expected value statistics are structured to compare different candidate systems without requiring a high level of certainty in the data. These two model types can be best seen in the types of performance metrics selected for this analysis.

6.1.3.1 Task Load Model Performance Metrics

A few overall objectives permeate the entire mission planning spectrum. The mission task list total completion time should be minimized. This should not be interpreted to mean that this objective was paramount. On the contrary, it should be considered along with the other objectives as a goal, but flexible to enable minimization of the other objectives.

In addition, a primary criterion used to assess the value of a cooperative HRT over the standard two-human crew from a task load model perspective is the reduction in time that the human crew must wait for the robotic agents to perform their subtasks. This reduction represents a specialization of the human activities and a check on the efficiency of the developed schedule (how efficiently the humans are used as a resource).

The *a priori* nature of the designer's perspective taken for this analysis facilitated selection of a work efficiency metric. Schreckenghost's work efficiency index [83] was defined to be the ratio of productive time to overhead time that occurs in an agents' task schedule. This can be viewed as the efficiency of resource use - whether an agent actively contributes to task completion or stands idle waiting for a precedent-constrained task to be finished.

Since it is desirable to maximize a work efficiency index, the inverse was used in this analysis to enable a minimization objective function. For the human agents, the ratio of human inactive time to total human involvement time would represent the efficiency with which the human agents were utilized as a resource. This metric was applied to each of the two types of crew agents in each team pairing. For the objective functions used in the MOGA, these two metrics are represented by minimizing the fraction of human inactive time to total human involvement time, and minimizing the fraction of robot inactive time to total robot involvement time.

The scenario perspective adopted for this analysis was both human-centered and robot-centered. Although preference was given in the task allocation scheme to humans (assigning tasks to robots if they were equipped), the performance metric analysis did not have a preferred resource. Both the human agents and robot agents were treated equally. Both the total time that human agents were active and the total time that robotic agents were active were objectives to be minimized. By minimizing each agents' active time, the analysis sought to make each a more specialized resource.

Autonomy levels were set for each team pairing. No sliding autonomy was considered because this applies to real-time mission scenarios rather than *a priori* mission scenarios. Supervisory control of the robots was assumed, although the team pairings would be equally valid for autonomous robotic systems. The task performance time would be expected to change for fully autonomous systems, but as this data was merely representative for this research, this effect was neglected. If the autonomy of each of the teams and their component agents had been specified, it would be beneficial to utilize comparative autonomy metrics like those discussed in section 2.1.1. Similarly, a communication architecture could be specified in a more comprehensive performance analysis. For this research, some robots were given capacity to perform the verbal task primitive, enabling them to take part in a wider range of subtasks. Other consequences of a communication architecture were not considered in this experiment.

To represent human physical workload, a modified Cooper-Harper scale was utilized (as suggested in [70]). According to the modified Cooper-Harper scale, workload ratings of 3 and below correspond to small and inconsequential errors made during an acceptable level of effort. A rating of 1 is deemed very easy, when an operator's mental effort is minimal and the desired performance is easily obtainable. A workload rating of 2 is deemed easy, when an operator' mental effort is low and desired performance is attainable. A rating of 3 corresponds to a fair to mid difficulty level, when an operator's required mental effort is acceptable to reach adequate system performance.

To develop performance metric data for each of the 82 unique team combinations and their corresponding schedules, values were assigned based on the number of physical subtasks the human agents were required to perform. Corresponding to the number of task primitive types that each agent performed, if an agent performed fewer than 3 different types of tasks, they were rated with a physical workload value of 1. Less than 5 types of primitives corresponded to a workload value of 2, and more than 5 types of primitives corresponded to a workload value of 3. To add more variability to the performance metric scores for the unique team combinations, a value between zero and 0.3 was randomly added to the scores such that, for example, the workload values of 3 were randomly selected on the range between 3.0-3.3, inclusive.

Mental workload for an intra-vehicular astronaut controlling or supervising a robotic agent was assessed depending on whether the robotic agent had the capability to provide visuals to the controller (one of the task primitive types used to characterize each of the agents, as discussed in section 6.1.2). In this usage, an operator's situational awareness of the robot's work environment directly influenced mental workload. Using the modified Cooper-Harper scale for unmanned vehicles (as suggested in [70] to gage mental workload during off-site supervisory control), a workload rating value of 2 corresponds to an operator having an acceptable display of the vehicle's environment. The display may have minor issues, but they do not interfere with the operator's performance. A workload rating value of 4 corresponds to an operator having insufficient information about the vehicle's environment to enable decision making.

In this research's performance metric analysis, if the robot supplied a visual of its workspace, the mental workload was rated at a 2. If no visual of the workspace was provided by the robot, the workload rating was a 4. If no robots were used in a specific team pairing, then the IVA astronauts mental workload was rated at 0 (no supervisory control was required). To add more variability to the performance metric scores, the workload values had an additional randomized value up to 0.3 added to the final score such that a rating of 4 was assigned on the interval from 4.0 to 4.3, inclusive.

6.1.3.2 Expected Value Statistics Performance Metrics

Performance metrics in this category sought to compare the relative performance of each candidate system to a reference. This facilitated use of metrics for which exact performance data was unavailable. Should the robotic agents of candidate system be more thoroughly specified in future analysis, their specific performance data would allow selection of different performance metrics. For this research, however, these metrics allowed consideration of a wider selection of team performance characteristics.

It would be intuitive to maximize the autonomous navigation speed of each of the agents (metric described and used in Tunstel [96]). Higher traverse speeds would enable a larger range of territory to be covered in a shorter interval of time. Since the MOGA for this experiment was being built as a minimization function, this performance metric was represented as an expected value statistic, gaging each of the candidate robotic systems against a fully functional reference system. If both of the potential robotic agents in a team pair were capable of traversal, the autonomous navigation speed rating of the team pair was given a value of 10. If either of the two potential robotic agents in a team pair were capable of traversal, the autonomous navigation speed rating of the team pair was given a value of 50. If either of the two potential robotic agents in a team were present and active (if either was capable of at least 1 task primitive type) but both were incapable of independent movement, the autonomous navigation speed was given a rating value of 150. If there were no robots in a given team combination, then the human pair were given a rating value of 10, equivalent to the fully functional robotic reference system. To add more variability to the performance metric scores, the navigation speed rating values had an additional randomized value up to 20% added to the final score.

Similarly, it would be intuitive to seek to maximize the autonomous traverse distance of each agent (a measure of longevity of the agent, with use demonstrated in Tunstel [96]). This task would require not only the ability to translate (as measured by the autonomous navigation speed metric), but would also require the capacity to visually inspect the terrain during this travel. The reference system for this performance metric was two fully functioning robotic agents, capable of both traversal and visual inspection. A rating value of 1 was given to this type of system.

If only one robot was capable of traversal and visual inspection, this robot earned a rating of 10, equivalent to the fully functioning candidate system. If only one robot was capable of traversal and visual inspection but both were capable of traversal, a rating value of 50 was earned to denote the added workload required by the one robot to provide visualization for both robotic agents. If only one robot was capable of traversal, but this robot was not capable of visual inspection, the team earned a rating value of 50. If either robot was present and active (if either was capable of at least one task primitive type) but both were incapable of traversal and visual inspection, the team earned a rating value of 150. As with the previous metric, if there were no robots active for a candidate team (i.e. a human-only team), an autonomous traverse distance rating value of 10 was used. To add more variability to the autonomous traverse distance score, the values had an additional randomized value up to 20% added to the final score.

Tunstel's approachability performance metric [96] was instrument-dependent. It related the number of instrument targets a given robotic system could reach, relative to the total distance traveled to reach those targets and the total amount of time required. For this analysis, the reference system was capable of not only translating and visual inspection, but was also required to be carrying an on-board tool for data acquisition of its environment.

If a robotic system was capable of each of these three task primitive types, it earned an approachability rating value of 10. If a robotic system was capable of traversal and visual inspection but did not carry an on board tool, it earned a rating value of 30. While this might seem a generous rating, it was anticipated that the robotic agent would find and locate targets. A human agent would then arrive at the specified target and use the instrument. If neither robot was capable of traversal, visual inspection, or carrying a tool, an approachability rating value of 100 was used to mark the team as incapable according to the approachability metric. As with the previous metric, if there were no robots active for a candidate team, an approachability rating value of 10 was used indicating equivalent performance to the robotic reference system. To add more variability to the approachability rating, the values had an additional randomized value up to 20% added to the final score.

The error-rate for each agent on a candidate team was determined as a function of the number of task primitive types each agent was required to perform during each subtask. If an agent performed three or fewer task primitives in a given subtask, its error rating for that subtask was 0.001 times the number of task primitive types required. If an agent performed six or fewer task primitives, the error rate was 0.01

times the number of task primitives required. If an agent performed more than six types of task primitives in a given subtask, the error rate was 0.1. The increase in error rate corresponds to the notion that if an agent performs a wider selection of tasks, this increases the chance that the agent will make a mistake. The final error rating for each agent was the summation over all of the subtasks that the agent performed for each team. This error rating metric was a stand-in for more substantial technical data that would be available if candidate robotic systems were specified for this method. The final error value then corresponds to the number of potential errors that each agent would be anticipated to make.

The final performance metric used for this large HRT configuration selection problem analysis was designed to be a cost function to penalize over-qualified teams. For example, if two teams are equally capable of performing the entire mission scenario, then the least capable of these teams would be the preferred team. Each added primitive capability of the more qualified team represents an unused and redundant skill.

Each agent on each team was assessed to determine the number of primitives it was capable of performing, and a cost was applied for each. The total number of primitives for the two human agents were multiplied by a factor of 10 for numerical relevance in comparison to the other objective function values. Each of the robotic agents' cost was computed in the same way with the addition of up to 20% added cost to denote that, in general, designing for robotic systems has a higher fabrication cost than designing for humans. Each overall team's cost value was the summation of each of the four agents' individual costs.

6.2 Large HRT Configuration Selection Experiment

There were several steps to setting up the experiment. All of the large HRT problem-specific information was calculated and stored in a database. This database would be referenced several times during the MOGA simulation. The fitness function used by the MOGA translated the design variables into potential team combinations, checked them against the database, and returned their corresponding information. The MOGA parameters were specified for each run. Each of these areas will be discussed in more detail, and the results of this experiment are discussed in the following section.

6.2.1 Inputs: Database Development

Several databases were created prior to running the simulations of this experiment. The 82 unique team combinations (defined solely by the task primitives each of the four possible agents were capable of performing) were sent through the task allocation scheme. The result was a task performance schedule for each of the 82 team combinations that specified the order of task performance and which agents were required for each of the subtasks. Each of these team combinations and their corresponding schedules were evaluated across all of the performance metrics specified in section 6.1.3. All of this information (agent task primitive capabilities for each team, team schedules, and team performance data) was stored in a Matlab data file, to be used as a reference database for the experiment simulation.

It was noted that although there were 82 unique team combinations, there were not 82 unique schedules generated for the database. Due to the simplified task allocation scheme used in this research, several of the team combinations resulted in the same task allocation and performance schedule. This can be seen in multiple ways. First, if two distinct teams perform the schedule in the same way, then any additional capabilities or skill sets of one of the teams over the other represent unused capabilities. The more equipped and advanced robotic agents from these scenarios did not result in schedule or overall team performance improvements. These robotic agents were under-utilized in the schedules.

Several of the performance metrics implemented were designed specifically to catch this type of under-utilization and penalize the team score (approachability, workload, autonomous traverse distance, cost, etc.). Only those metrics that relied solely on data from the team performance schedule (total mission completion time, agent active time, etc.) would show no numerical difference between the different team combinations.

6.2.2 Defining the MOGA Fitness Function

Most of the operations required for this simulation were grouped into the fitness function of the multi-objective genetic algorithm (implemented in Matlab). A detailed description of the data flow and functions written for the HRT fitness function can be found in Appendix B. Fundamentally, the fitness function discretized the design variables from each MOGA generation into potential team combinations. These combinations were either present in the database, or the team pairing was identified as not one of the feasible combinations.

The teams represented by each of the design variables were assessed for their objective function performance (according to the performance metrics), and the mission constraints were applied to each design point. Each of these modified fitness values (reflecting both objective function values and feasibility of the design point) were fed back into the MOGA to be used in generating the next population of design points.

6.2.3 Specifying MOGA Parameters

When running the MOGA for this set of simulations, the built-in MOGA control options were altered. The first instantiation of the MOGA in this simulation had been intended to run through 100 generations and then stop the search process (with an elite count of 15 individuals, average spread in the Pareto front of 1e-02, 200 individual population size, crossover fraction of 0.8, and the adaptive mutation algorithm set to 0.2). This would provide a large population of potential data points to initialize the objective reduction algorithm.

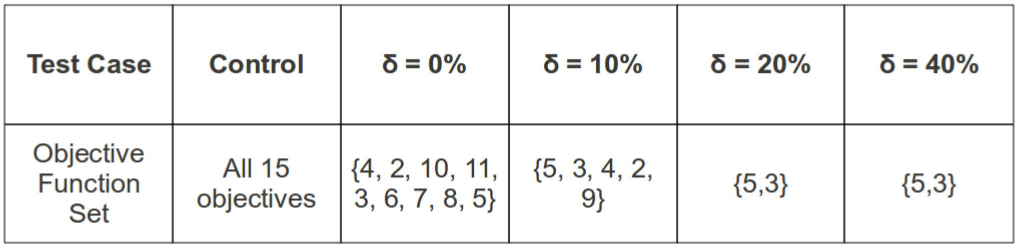
Each of the reduced objective sets (with a δ-error of 0%, 10%, 20%, and 40%) were run through the MOGA. The options were selected for this set of experiments to enable the MOGA run through its entire search process and come up with its best Pareto front for each of the given problems. The MOGA was set to run for 400 generations in each of the problem instances, maintaining a population of 200 individuals, until either the maximum generation counter was reached or the average spread in the Pareto front was on the order of 1e-06 (tolerance to determine that the Pareto front has converged), with an elite count of 15 individuals, average spread in the Pareto front of 1e-02, 200 individual population size, crossover fraction of 0.8, and the adaptive mutation algorithm was set to 0.2.

6.3 Results and Discussion

6.3.1 Objective Reduction for the HRT Configuration Selection Problem

Brockhoff's δ-MOSS algorithm for objective reduction was utilized in this experiment to reduce the original 15 objective function set. The input to the objective reduction algorithm was the MOGA results from an initial 100 generation run. The δ-MOSS algorithm was then run four different times: with a δ-error of 0%, 10%, 20%, and 40%. The results of this algorithm are tabulated below, in Figure 6.1, and each of the objectives listed are ordered in terms of importance as specified by the δ-MOSS algorithm.

As anticipated, the δ-MOSS algorithm proved capable of reducing the size of the objective set for the large HRT demonstration problem. The algorithm reduced the required number of objectives from 15 down to 9 in the 0%-error case - a substantial improvement in the complexity of the problem without introducing an error into the underlying dominance structure of the problem. In other words, the 15 performance metrics contained redundant information about the underlying relations between the objectives. Even though each of the performance metrics measured a



Test Case	Control	δ = 0%	δ = 10%	δ = 20%	δ = 40%
Objective Function Set	All 15 objectives	{4, 2, 10, 11, 3, 6, 7, 8, 5}	{5, 3, 4, 2, 9}	{5,3}	{5,3}

Figure 6.1: Large HRT Configuration Selection Problem: Reduced Objective Sets Resulting from the δ-MOSS Algorithm

different quantity, 6 of them could be removed from consideration without altering the underlying structure of the problem.

Figure 6.1 contains the results for each of the four δ-error test cases. Recall that the ordering of the objective functions in this figure corresponds to the ranking assigned by the δ-MOSS algorithm according to the quantity of pairwise relations represented in the data. Allowing a 10% δ-error in the underlying dominance structure of the problem allowed the objective set to be further reduced down to five objectives. Increasing the error to 20%, the objective set was shrunk to include only two objectives. Increasing the error tolerance beyond 20% did not further reduce the size of the objective function set. The same two objectives were required to preserve the underlying dominance structure in the last two test cases.

The third and fifth objectives (EVA day length and robot day length, respectively) were deemed important for preserving the dominance structure in all of the test cases, though their importance in contributing problem information differed (as represented in Figure 6.1 by the objective function numbering). It is interesting to note that these two performance metrics were selected - they represent the crux of the decision-making within a HRT team: how many tasks and how much time to allocate to the human portion of the crew and to the robot portion of the crew to best optimize agent usage.

In both the $\delta = 0\%$ and the $\delta = 10\%$ test cases, two other objectives were prominently listed in the reduced objective set: the fourth and second objectives (overall mission time and the cost of each team, respectively). Put together, these four objectives identify most of the design trade-offs within this instance of the large-scale HRT configuration selection problem. These objectives balance which agents are utilized within the mission scenario with preference for the least capable of the agents (to reduce the cost of the overall team). The most equipped team (2 human crew with 2 Robonaut-like robotic agents) would have the potential to perform the overall mission fastest with an equal split in workload levels between the human and robotic agents, but this crew would also be the most costly in terms of built-in capabilities.

The fifth objective listed in the $\delta = 10\%$ test case's reduced set (the ninth objective - autonomous navigation speed) was only selected in that test case. Combined with the four objectives just discussed, this additional objective included consideration of how well the robotic agents were able to perform their tasks, rather than merely considering the completion time required. This performance metric allowed differentiation between the different kinds of robotic agents - not only would the cost of a robotic agent's capabilities be considered, but the ability of those skills to contribute to performance efficiency would be included in judging the design space.

Autonomous navigation speed was not one of the metrics selected in the $\delta = 0\%$ test case. Instead, other measures of performance efficiency were selected. Objectives 10 and 11 (approachability and autonomous traverse distance) helped characterize how well robotic agents performed the survey-task portions of their scripted missions. Objectives 7 and 8 (human physical and mental workload) depicted how hard the humans had to work to complete their assigned portion of the task list. Objective 6 relates the work efficiency index for the robotic agents, indicating the fraction of the entire mission that the robotic agents were involved.

Considering the performance metrics selected in the $\delta = 0\%$ test case, the metrics represented mission and agent time, and the efficiency of each agent's task performance. All of these metrics combined represented a non-redundant set of problem-specific information that wholly characterized the objective space.

It is interesting to note that five of the performance metrics included for consideration in the original problem description (objectives 1, 12, 13, 14, and 15 - human work efficiency index, and the error rates for each of the four agents (based on variety of task primitives performed in each team scenario), respectively) were deemed wholly redundant by the δ-MOSS algorithm and were never selected for placement in the reduced basis.

From an initial analysis of the 15 original performance metrics, it would not have been possible to determine which (if any) of the metrics contained redundant information. These five metrics had been included because they contained additional information about how well each of the agents performed their respective missions on each of the 82 teams. None of these metrics correlate to the same primary data as the other metrics - they all appear to calculate independent values. The error rate performance metrics sum not only each agent's skill set, but the usage across the task allocation and developed schedules. The human work efficiency index differentiates between human involvement in the mission scenario and the human actively working on a task (it records human inactive time as an inefficient use of humans as a resource).

It should be noted, however, that the selection or rejection of performance metrics for this demonstration problem directly correlates to problem-specific data. For other large scale HRT configuration selection problems, it would be faulty to select performance metrics based on their selection in this problem. Different candidate HRTs will have different pairwise relations across all of the performance metrics, and a complete picture should be sought for each new problem instantiation.

6.3.2 Pareto Results for the HRT Demonstration Problem

This experiment followed the analysis steps laid out in the methodology discussed in Chapter 3. The reduced objective sets were used to run the MOGA 5 times for statistical relevance. Each run sought to reach convergence of the Pareto set. The output from all of the runs were collated and sent a final iteration through the continuous update algorithm to verify that they represented a set of nondominated solutions. The results from this final analysis can be seen in Figure 6.2.

Figure 6.2 displays each of the 82 unique teams that were feasible solutions for

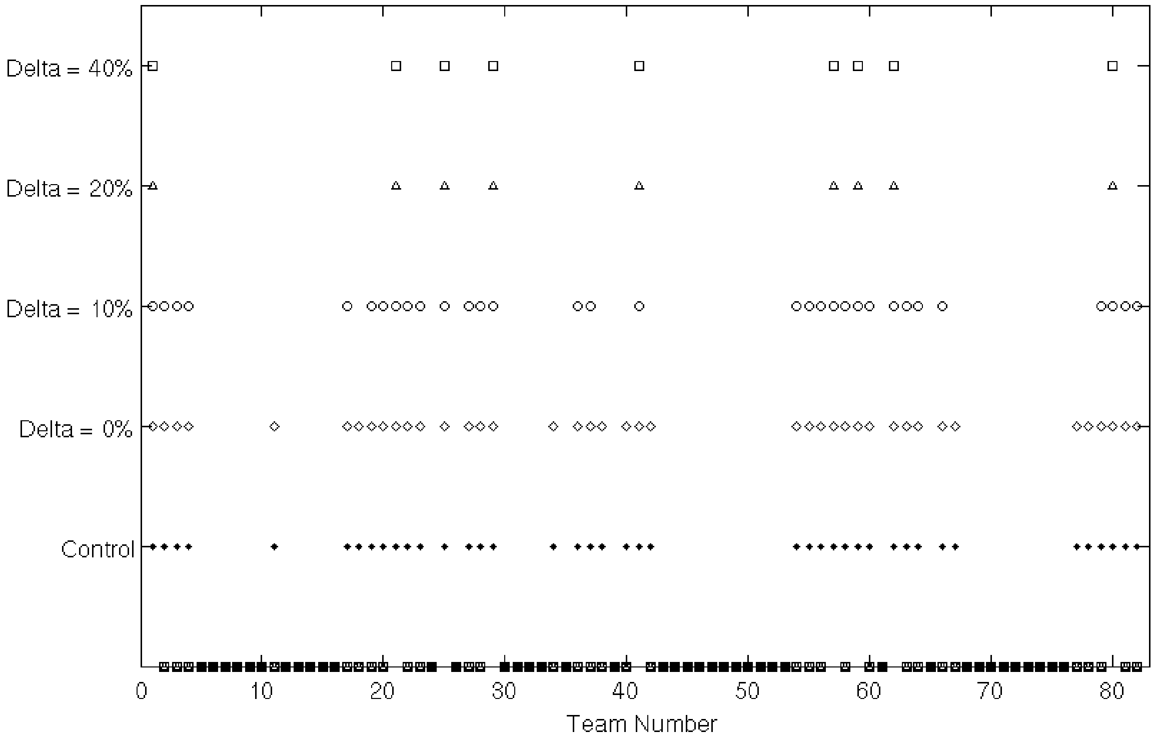


Figure 6.2: Teams Selected for Large HRT Demonstration Problem in Each of Five Test Cases

the large HRT demonstration problem on the x-axis. On the y-axis are the five test cases that were run (vertically offset for visibility only - there is no significance to height on the y-axis). There are several interesting trends to notice about the Pareto sets from this Figure. First, the Pareto sets resulting from the Control test case and the $\delta = 0\%$ test case are identical even though there was a substantial difference in the performance metrics used to generate them (6 additional performance metrics were used in the Control test case that were deemed redundant in the $\delta = 0\%$ test case by the δ-MOSS algorithm). This result justifies the use of the reduced objective function set - with a δ-error of 0%, zero change was incurred in the underlying dominance structure of the problem. Both of these Pareto fronts demonstrate that the MOGA found 41 unique candidate teams that performed no better than each other. The mission designer's decision space was reduced from 82 teams measured across 15 objectives in the Control test case down to 41 teams measured across 9 objectives.

Recall that allowing a 10% error in the underlying dominance structure of the problem reduced the objective set from the 9 non-redundant objectives from the 0% test case down to five objectives. This smaller set of criteria from which the solutions were judged helped winnow down the number of solutions in the Pareto set for the 10% test case to 32 teams. Including a 20% error brought that solution set size down to 9 teams in the final Pareto set. The final Pareto set for both the 20% test case and the 40% test case were identical. With the two performance metrics in the reduced objective set, these nine solutions were identified as better solutions than the other 73 candidate teams. Expanding the allowed δ-error in the

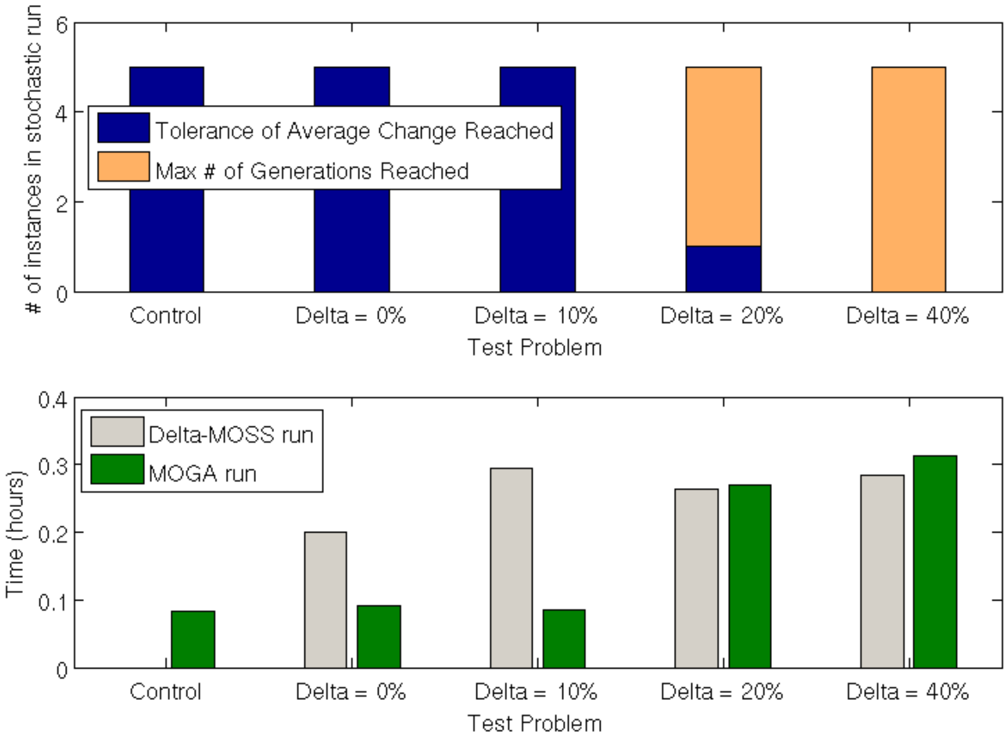


Figure 6.3: Termination Reason and Run Time for Large HRT Demonstration Problem

dominance structure of the underlying problem allowed further down-selection of the teams represented in the Pareto front, easing a mission designer's final selection of a team configuration.

Figure 6.3 demonstrates the convergence success for each of the test problems across each of the 5 stochastic runs through the MOGA. The control case, $\delta = 0\%$, and $\delta = 10\%$ test cases all converged 100% of the time. The MOGA was able to locate the Pareto front even with the larger number of competing objectives. In the $\delta = 20\%$ test case, this convergence rate dropped to 20% and in the $\delta = 40\%$ test case this fell to 0%. In these last two test cases, the MOGA searched along two strongly conflicting objectives for the Pareto front. However, the fitness function had been written to be insensitive to small decimal changes between generations of design variables (a low level of rounding was employed). This meant that the MOGA did not reach the small average tolerance of Pareto spread that it was searching for as the criteria for convergence.

As had been the case with the knapsack test problem instances from Chapter 4, the computational time required for the δ-MOSS algorithm was the primary disadvantage of this method. As seen in Figure 6.3, the δ-MOSS algorithm required more time than the control test case's MOGA, and the MOGA run time for each of the δ test problems increased in time after a δ-error of 20% was included. In comparison, the control test problem's MOGA required less than 10 minutes to complete. As had been concluded from the knapsack test problems, if computational time is a limiting factor for future performance analysis, this methodology would not be advantageous.

However, if computational time is not an issue for *a priori* analysis, a significant reduction in problem complexity can be achieved by utilizing the methodology proposed in this dissertation. Taking the $\delta = 20\%$ test case as example, a run time equivalent to four times that of the control case reduced the problem from a consideration of 41 teams in the final control Pareto set compared across 15 objectives to a consideration of 9 teams compared across 2 objectives - a substantial reduction.

A final analysis on the solution quality of the Pareto sets resulting from each of these five test cases was performed, utilizing the hypervolume quality metric (see section 3.2.4). These results are tabulated in Figure 6.4.

There are several important trends to note from the data represented in Figure

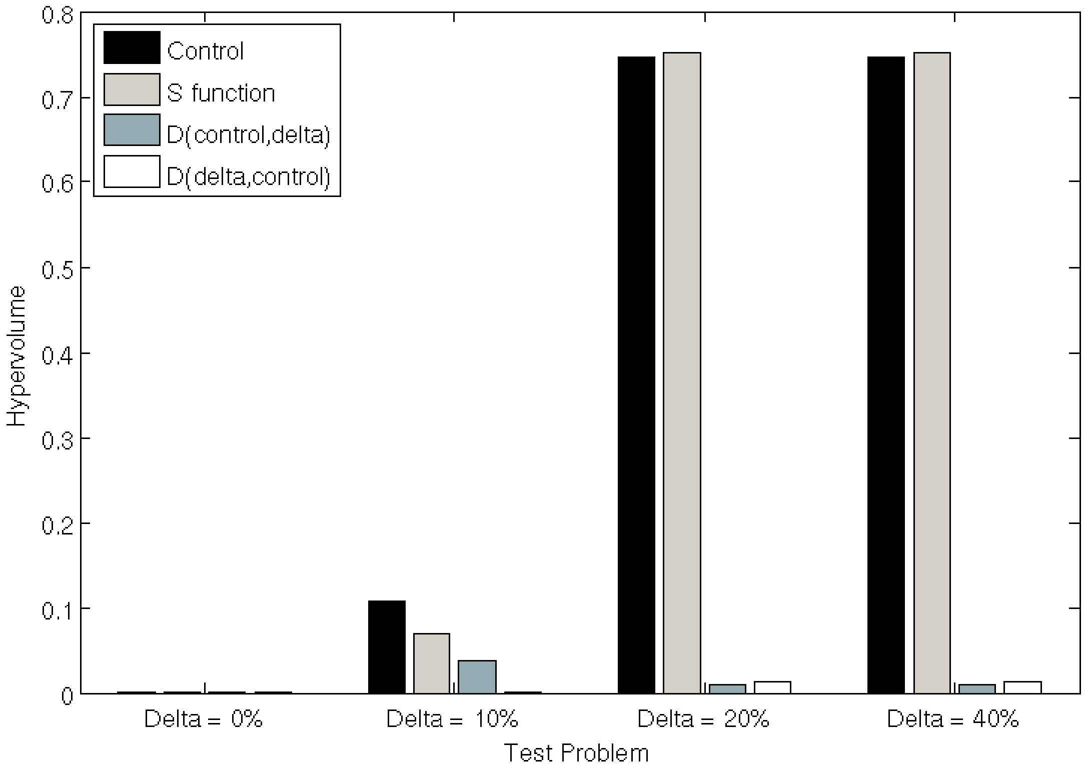


Figure 6.4: Hypervolume Indicator Quality Metric for Each Test Case of Large HRT Demonstration Problem

6.4. Recalling that the S quality metric reflects absolute volume coverage of the Pareto front, it can be seen for each of the four test problems that the reduced objective set resulted in a Pareto front with nearly identical volume coverage. Each Pareto front generated from this analysis represented remarkably good coverage across all of the objectives.

Recall that the D quality metric was the relative volume coverage used to compare two different Pareto sets. In this large HRT problem, the control was compared across each of the four test cases with different δ-error values. In the $\delta = 10\%$ test case, the Pareto volume coverage of the control set was approximately 5% better than the Pareto volume coverage resulting from the reduced objective set. This result is consistent with those seen in section 3.2.4 for the knapsack test problem.

However, the results from the other three test problems in the large HRT problem are impressive. A fraction of 1% was calculated for the relative volume coverage both comparing control to the δ-Pareto set, and δ-Pareto compared to the control. There were no portions of the Pareto front that were covered by one set and not by the other. This provides the analytical support for the assertion that the Pareto sets resulting from the reduced objective sets not only provide a good approximation of the control Pareto set (in general), but (in this case) provided *the exact same coverage of the Pareto front.* A 1% error in solution quality would be introduced if a mission designer chose to use the Pareto set generated by the 2-objective $\delta = 20\%$ and 40% test cases instead of the 15-objective control case.

This result is the final justification that was sought from running the large

HRT demonstration problem - application of the methodology proposed in this dissertation provided the rigorous performance analysis that had been missing from previous research in the literature. The δ-MOSS algorithm provided objective, analytical reasoning for selecting some performance metrics and deeming others redundant. Rather than running a comprehensive performance analysis with 15 objectives, a researcher could objectively and conclusively run the performance analysis with fewer objectives at any of the given error-tolerances and achieve very similar results.

6.3.3 Team Configuration Analysis

A closer examination of the best performing teams from the 20% and 40% (those depicted in the Pareto sets in Figure 6.2) test cases will be illuminating. Team 1 represented the standard two-human crew from the original NASA mission, without any additional robotic agents.

The speed of the Robonaut-like robots on the Pareto teams for these test cases represented an interesting trade-off in design parameters. Team 41 contained the standard two human crew and two Robonaut-like robots that performed tasks at 1/2 the speed of the human crew. Team 62 contained the standard two human crew supplemented by one Robonaut-like robot (equivalently capable to a third human crew member) that required three times the amount of time to perform a task as its corresponding human crew. The extended time required for robot performance was traded with cost for the extra skill capabilities for the team such that the 1/2 speed robots were more worth the cost than the 1/3 speed robot in the overall team picture.

Team 21 has the standard two human crew supplemented by one Robonaut-like robot (equivalently capable to a third human crew member) and a fourth agent with the skill set of an astronaut-assistant rover with no specialized tools or skills (capable of independent translation, load carry, site survey, force-push, and force-pull). Team 59 represents the same team agents as team 21 with the restriction that the robotic agents required twice as long to perform tasks as their human counterparts. It is interesting to note that both of these teams were included in the final Pareto front, and also that the same team configuration that performed tasks at 1/3 the rate of the human crew was not present in this final set. It is clear that the trade-off between slower robot (less costly) and mission completion time could allow the two different robot speeds for the teams, but that the 1/3 rate robot was too slow and brought down overall team mission performance.

Teams 29 and 57 involved a variation on the Robonaut-like robot with all skills except the ability to independently initiate communication. Team 29 supplemented this robot with the two human crew and an independently mobile robot that could provide an additional camera and visual perspective (verbal, visual, and translation primitives). Team 57's Robonaut-like robot required twice as long to perform its tasks, and the team varied the second robot agent by removing the translation primitive. The fourth agent on this team was capable of providing a visual perspective and communicating its observations to the other agents, but required an external agent to place it at a desired location. Both of the teams' robotic agents contained between them all of the primitive skills needed to complete the full set of tasks. The two teams vary in the details of how the two robotic agents would split the tasks.

Team 80 contained the standard two human crew, one Robonaut-like robot (equipped as a third human), and one robotic agent that could provide an additional visual perspective but required an external agent to place it at a desired location. Both of the robotic agents on this team moved at 1/3 the speed of their human counterparts.

Team 25 was the only configuration in the final Pareto set to involve a different skill set in the Robonaut-like robotic agent. The primary robotic agent on this team was fully capable except that it was not able to use a specialized tool. This would be a version of Robonaut limited by its end effectors (or robotic hands) available for a mission. It was capable of gripping handrails (EVA interfaces) and using a pinch grasp for delicate holds, but would be incapable of using other specialized tools like drills. This team was supplemented by a fourth agent that provided an additional visual perspective and was capable of independent mobility.

Stepping back from the details of the candidate teams selected for the final Pareto sets of the $\delta = 20\%$ and 40% test cases, there are several overarching trends to be noted. In the representative best-performing teams, there were fundamentally only three different types of robotic agents: a Robonaut-like agent with the skill set of a human (or only one primitive off from this robot), an astronaut-assist robot capable of site survey, independent mobility, and carrying a load between desired locations, and an intelligent camera (in some instances it had independent mobility, in other cases it required another agent to place it). The teams in the final Pareto set contained combinations of these robots performing at different speeds to supplement the standard two human crew. It was the robotic agents that could substantially offset the human crew's workload that were most useful to the overall team performance. However, it is intriguing to note that fully equipped robots were not necessary for this balance to be obtained.

Why did other variations of robotic agents not make the cut into the final best-performing set? This can be traced back to the task allocation and scheduling schema utilized for this demonstration HRT configuration selection problem. During the task allocation process, if a robotic agent was capable of performing all of the primitives within a subtask, then that subtask was allocated to the robot. Otherwise, it was allocated to the human agents. Robotic agents with sparser skill sets were not capable of performing a sufficient percentage of a subtask to be tasked with it. In other words, candidate teams that had these sparser skilled robotic agents did not use them sufficiently to offset the cost of including the agent in the team.

It is possible to assess the nine different types of task primitives and use this information to determine the configurations of robotic agents that provide sufficient utility (based on the assumptions built into this demonstration problem) to offset the human crew's workload enough to justify the added cost of the robot. The unique skill combination of visual inspection and communication initiative was the least-skilled robotic agent in the Pareto set. The next combination of task primitives included translation for independent mobility. These two different combinations represent a category of survey robots that aided the human crew by exploring additional territory without the human crew needing to traverse.

From this base skill level, the added capability of load carrying to transport objects was greatly useful for offloading the workload of the human crew. From this level, however, the robotic agents were not sufficiently skilled enough without several more primitive type-capabilities including force-push, force-pull, and handrail grip, which combined to allow a survey robot to independently pick up and put down the objects it had been carrying.

In the final Pareto set, there is a very clear trade-off between robot skill level and robot speed. The more equipped robotic agents frequently were designated to operate at a slower speed than their human counterparts. This is the reason that some team combinations only existed in the final basis with slower versions of the robots and why their faster versions did not make the basis.

This correlation between robotic agent skill level, task performance speed, and overall cost represents an intriguing guiding principle for future design of robotic agents and of optimized human and robot teams.

6.3.4 Discussion of Reproducibility

A final question that should be asked in this analysis is what the variation in the composition of the reduced objective sets say about the reproducibility of these sets for selecting performance metrics. In the case where two performance metrics contain equivalent problem data and represent the same pairwise dominance relations, the first encountered in the δ-MOSS algorithm's iterations will be selected for placement into the reduced objective set. In the next iteration, therefore, all of the relations contained in the second equivalent performance metric will already be represented in the basis. This second performance metric will not be selected to be included in the basis. While the initial input ordering of performance metrics will obviously have an affect on those selected for the reduced set, the pairwise dominance relations represented by the final reduced set will remain the same. In other words, a variation in the performance metrics used in an overall analysis, if varied by the δ-MOSS algorithm, will not have an effect on the resulting Pareto solution sets.

In a sense, all of the problem detail can be represented by these candidate solutions. The algorithm determines which objective functions provide the most information for the given candidate solutions. When a set of poor candidate solutions were input into the δ-MOSS algorithm, a different set of objective functions was deemed necessary to reflect the underlying dominance structure than when a set of mixed good and poor candidate solutions were input into the algorithm.

From this perspective, it would be expected that the quality of the solutions would cause variation in the reduced objective set. When applying the δ-MOSS algorithm, the reduced set of objectives will be selected based on non-arbitrary, problem-dependent information. No preference information will be needed. A large number of candidate objectives can be used to canvas the decision space, or a smaller set could be used. Either way, the algorithm will isolate the underlying dominance structure of the problem domain and identify those performance metrics that are most needed to characterize the decision space. The objective reduction algorithm has proved itself an immensely valuable tool for identifying the critical information for various problem types and across different domains.

6.3.5 Discussion of Complexity Reduction in Pareto Decision Space

It has been demonstrated that the Pareto solution sets generated by utilizing this dissertation's proposed methodology are approximately (if not exactly) equivalent to using a large, unreduced objective set. It has been analytically demonstrated that large, complex, multi-objective optimization problems can be generically, objectively, and conclusively reduced by application of this methodology.

How much does this methodology aid a mission designer's decision making? There has been significant reduction in the complexity of the mission designer's decision space. Where previously the mission designer had a design space detailed by 82 unique teams compared across 15 performance metrics, this methodology reduces the design space to 41 unique teams compared across 9 performance metrics without introducing an error in the underlying problem. This is a substantial improvement for the mission designer. Including a δ-error further reduces the designer's decision space to consideration of 9 unique teams compared across 2 performance metrics.

A further iteration on this experiment could apply a more complex and selective task allocation schema that would further differentiate the schedules and, therefore, the 82 team options. It would be anticipated that this would further reduce the number of candidate team options represented by the Pareto fronts in all of the test cases, and further reduce the options that a mission designer must choose from.

This experiment was able to significantly reduced the designer's decision space. Additionally, the designer was analytically assured that the results from this reduced decision space were at least as good as a solution selected from the design space represented by the control's solution set.

6.4 Summary

This experiment sought to demonstrate the methodology proposed by this dissertation on a large-scale HRT configuration selection problem. The other experiments from this dissertation had suggested that the application of the methodology to this design space would facilitate significant reduction in the complexity of the over-constrained, multi-objective optimization problem, and that it would provide the rigorous objective reasoning to down-select from a large set of performance metrics to the few significant ones for analysis.

82 unique teams were proposed as candidate solutions and were compared across 15 performance metrics. For a mission designer, this is a very large decision space to consider. In the research described in Chapter 2, previous mission designers would have arbitrarily selected a small set of performance metrics to use in an overall team performance analysis. The lack of rigor in this type of analysis meant that comparison of results between different researchers, platforms, and team pairings was virtually impossible.

Applying the δ-MOSS objective reduction algorithm to this problem immediately (and rigorously) reduced the performance metric set from 15 down to 2 performance metrics. Although there was some variability in the reduced set depending on the tolerable error in the underlying problem structure, this significantly reduced the decision maker's design space. This by itself was a significant result. This research's methodology, however, took the analysis one step further to assess the Pareto solution sets that resulted from the reduced objective sets. It was this final analysis step that reduced the 82 unique teams down to the 9 best teams.

The methodology proposed in this dissertation research has been applied to three different application realms and has proved beneficial in all three. This methodology has wide utility in reducing the complexity of large-scale over-constrained, multi-objective optimization problems. It would be anticipated that the methodology will provide rigorous analysis on many future applications.

Chapter 7

Conclusions and Future Work

The diversity of robotic technology available creates a multitude of new opportunities in task performance. Utilizing the new capabilities for hardware, software, sensors and system integration, and communication architecture could lead to a greater level of mission diversity, and facilitate cooperative human and robotic team scenarios that had previously been impossible.

Future mission designers will have a large heterogeneous group of distinct agents (both human and robotic) from which to select the most productive or efficient team members. Team members can be used in various combinations to better utilize their capabilities and skills to create more efficient and diversified operational teams. This involves allocating tasks to provide the most benefit from the partnership, and creating the planning, scheduling, and software interfaces to support these efforts. An overall, objective team performance analysis would be beneficial to facilitate decisions in this design space. It would enable quantitative comparison between disparate teams and allow a designer to select the most effective agents to complete a series of tasks.

It is this final question that this research sought to answer. A methodology was developed to facilitate performance comparison amongst heterogeneous human and robot teams. This methodology made no assumptions about mission priorities, preferences, nor importance of performance criteria. Instead, it provided an objective, generic, quantitative method to reduce the complexity of the mission designer's decision space.

This type of overall mission analysis to differentiate team configurations could be an inordinately valuable tool for mission designers. The challenge then becomes creating a quantitative, overall model to measure a team's performance for a generic mission, to enable broad use of the analysis tool (and remove the need for mission-specific models). To facilitate comparison between different analyses, being able to specify an objective set of criteria would be immensely valuable.

It is this last point that was the motivation for this dissertation research. A generic, objective methodology was developed to aid in determining a pseudo-optimal HRT configuration for a mission scenario. The HRT configuration selection problem was utilized as the application that motivated the problem and was used to assess the quality of solutions after the problem had been solved. The methodology, however, provided a much broader search of the design space between performance metrics and optimization models, facilitating evaluation of the design options.

7.1 Discussion of Important Results

This dissertation's proposed methodology was applied to three very different applications to demonstrate its utility and diversity, and to investigate how the additional dimensions of the large-scale HRT configuration selection problem alter the anticipated results. Using the conceptually simpler knapsack problem domain, the utility of this dissertation's methodology was demonstrated. The reduced objective set for the $\delta = 40\%$ test case from the 15 objective knapsack problem instance (which had 11 fewer objective functions than the control case) resulted in a Pareto solution set that was only 3% worse than the control according to the hypervolume quality metrics. That result is substantial. A drastic decrease in problem complexity was afforded at only a 3% decrease in solution quality. Considering the ease with which a mission designer could examine the resulting 4-objective reduced set for this test case, this research's methodology proved to be very beneficial in this application.

In the robotic reconnaissance case study, it was demonstrated that Rodriguez's [74] composite task score method for performance analysis was highly dependent on the performance metrics selected, and did not provide the objective analytical results that had been desired in the research. It proved insufficient for the rigorous heterogeneous performance analysis conducted as the first part of this case study.

To remedy this, the δ-MOSS objective reduction algorithm was utilized to reduce the performance metrics proposed by Tunstel [96] for the MERs to a subset that maintained the underlying dominance structure of the problem. In other words, the reduced set of objectives resulted in no change to the problem details and the final solution set. With zero error in the dominance structure, the six performance metrics used in this simplified analysis were reduced to two conflicting performance metrics. Four of the six performance metrics contained only redundant information and could be removed from analysis without affecting the problem and its solutions.

Furthermore, in the worst case from the test problems, the control Pareto set dominated just under 1% of the δ Pareto front. In other words, by using the

δ Pareto set for the 10%-error test case instead of the control Pareto set, a 1% decrease in the quality of the solution set resulted. This decrease in quality, however, afforded a decrease in the number of performance metrics from 6 down to 2 metrics. Additionally, this slight decrease in solution quality allowed the reduction from four candidate rovers down to two candidate rovers: the K10 red and Spirit.

The large HRT configuration selection demonstration problem experienced significant reduction in the complexity of the mission designer's decision space by application of this research's methodology. Where previously the mission designer had a design space detailed by 82 unique teams compared across 15 performance metrics, the methodology proposed in this dissertation reduced the design space to 9 unique teams compared across 2 performance metrics. This is a substantial improvement for the mission designer.

A further extension of this experiment could apply a more complex and selective task allocation schema that would further differentiate the schedules and, therefore, the 40 team options. It would be anticipated that this would further reduce the number of candidate team options represented by the Pareto front, and further reduce the options that a mission designer must choose from.

The Pareto solution quality results from three of the test problems in the large HRT problem were impressive. An exact value of zero was calculated for the relative volume coverage both comparing control to the δ-Pareto set, and δ-Pareto compared to the control. There were no portions of the Pareto front that were covered by one set and not by the other. This provides the analytical support for the assertion that the Pareto sets resulting from the reduced objective sets not only provide a good approximation of the control Pareto set (in general), but (in this case) provided *the exact same coverage of the Pareto front*. Zero error would be introduced if a mission designer chose to use the Pareto set generated by the 2-objective test cases instead of the 15 objective control case.

In the representative best-performing teams, there were fundamentally only three different types of robotic agents: a Robonaut-like agent with the skill set of a human (or only one primitive off from this robot), an astronaut-assist robot capable of site survey, independent mobility, and carrying a load between desired locations, and an intelligent camera (in some instances it had independent mobility, in other cases it required another agent to place it). The teams in the final Pareto set contained combinations of these robots performing at different speeds to supplement the standard two human crew. It was the robotic agents that could substantially offset the human crew's workload that were most useful to the overall team performance. Candidate teams that had these sparser skilled robotic agents did not use them sufficiently to offset the cost of including the agent in the team. However, it is intriguing to note that fully equipped robots were not necessary for this balance to be obtained.

In the final Pareto set, there is a very clear trade-off between robot skill level and robot speed. The more equipped robotic agents frequently were designated to operate at a slower speed than their human counterparts. This is the reason that some team combinations only existed in the final basis with slower versions of the robots and why their faster versions did not make the basis. This correlation between robotic agent skill level, task performance speed, and overall cost represents an intriguing guiding principle for future design of robotic agents and of optimized human and robot teams.

The methodology proposed by this dissertation does not prove its worth across the board. One very clear disadvantage to the methodology is the increase in computation time required to reach a Pareto set for a given design problem (in the worst case large HRT problem, the methodology required four times as much computational time as the control experiment). If computation time is a limiting factor for a set of analysis, using this dissertation's proposed methodology would be a hindrance to the analysis effort.

However, if computation time is not an issue, the merit of this research's proposed methodology has been clearly demonstrated. If a mission designer is not concerned with producing results as fast as possible, the Pareto solution sets resulting from this methodology have been demonstrated to have been at least equivalent, if not better, than the control case for each of the three application domains. The objective reduction algorithm has proved itself to be a powerful tool to reduce problem complexity by focusing the analysis on relative performance relations rather than on relative numbers. It is anticipated to be a valuable tool in larger human-robot team configuration selection problems. It has been analytically demonstrated that large, complex, multi-objective optimization problems can be generically, objectively, and conclusively reduced by application of this methodology.

7.2 Conclusions

This dissertation's research has provided a valuable tool for performance analysis of large, heterogeneous, human and robotic teams. A quantitative, generic, and objective methodology has been developed and demonstrated that will facilitate the type of analysis that must be conducted to enable comparison and selection of teams and individuals, and to find the most productive and efficient team members. This overall team performance analysis would be beneficial and would enable quantitative comparison between disparate teams for future mission designers.

This research developed a novel methodology to facilitate performance comparison amongst heterogeneous human and robot teams. This methodology made no assumptions about mission priorities, preferences, nor importance of performance criteria. Instead, it provided an objective, generic, quantitative method to reduce the complexity of the mission designer's decision space.

Given a set of agents and given a diverse set of potential metrics, this methodology allows a designer to rigorously choose a team configuration and task allocation within the team to satisfy one or more metrics in an optimal manner. A designer will not have to make *a priori* decisions about which are the critical metrics, nor which agents provide more utility to overall team performance. The underlying dominance structure of the complex problem will be assessed by the δ-MOSS objective reduction algorithm, and decisions about redundancy and similarity would be made based on these pairwise relations.

This type of methodology has not been done in the research field, and represents a valuable, unique contribution. The significant, unique components of this dissertation have been demonstrated:

- Decomposition of the HRT configuration selection problem into three distinct realms, decoupling their analysis: classification of agent and mission details, planning and scheduling, and selection of metrics and teams.
- Select performance metrics to preserve underlying dominance structure of a problem - this is a new approach for how to choose performance metrics.
- Demonstration of a method to compute overall performance evaluation that does not rely on aggregating multiple performance metrics into a single performance function.
- Create a generic, rigorous, objective, quantitative methodology for the HRT domain to both select the performance metrics to be used in an overall team performance analysis and to reduce the complexity of a mission designer's final decision space. This type of methodology is novel for the domain and a valuable contribution to further the use of human and robot teams.

It should be reiterated that the synthesis of the four state-of-the-art algorithms does not represent the contribution of this dissertation. This dissertation's contribution was facilitated by the synthesis of these algorithms. The unique contribution of this research was to create a generic, rigorous, quantitative, objective methodology to aid in the selection of performance metrics for overall team performance analysis and to reduce the complexity of a mission designer's final decision space.

Using the three components of the HRT configuration selection problem analogy described in section 1.2, this research lies on the plane defined by the performance metrics and optimization techniques. This allowed the substitution of the knapsack problem for the HRT configuration problem because the application domain was used to feed data sets into metrics.

In other words, the solution technique was not limited to a specific application domain. This methodology can be used for a much wider range of applications. Having a generic, objective methodology like this will be a necessary step to advancing the goal of using cooperative human and robot teams in future space activities.

7.3 Future Work

There are several elements of this research that could be improved with future work. Most notable has been the significant computation time required for the δ-MOSS algorithm. If the algorithm could be made more efficient by algorithmic improvement of code implementation to reduce the time required to run the algorithm on a large-scale problem, the method would be more useful and practical for a wider range of applications.

It would be interesting in future work to implement Brockhoff's exact δ-MOSS algorithm rather than the heuristic greedy approach used in this research. The implemented greedy approximation algorithm does not guarantee that the minimum objective set will be found - only a minimal set. The reason that the greedy algorithm had been applied was because Brockhoff had described that the exact algorithm failed with large complex problems [12]. If the large-scale HRT demonstration problem could be run with the exact algorithm, an analysis of the different reduced objective sets could be very illuminating for the underlying nature and relations between the performance metrics. In addition, an exact algorithm would be guaranteed to find the δ-minimum set. This would increase the uniqueness and reproducibility of the reduced objective sets.

As evidenced by several of the stochastic runs on each of the experiments, Matlab's multi-objective genetic algorithms toolbox struggled to reach convergence on some of the problems. The Matlab tool was utilized in this research both for its wide distribution (enabling other researchers to use it), and also for its ease of implementation. The interplay between the user defined fitness function and the MOGA itself could be made more efficient, or a real-time objective reduction method could be implemented to wheedle down the number of objectives during the MOGA runs themselves. Both of these options could improve the convergence of the MOGA.

Alternatively, an entirely different implementation of a MOGA could be utilized in future work. There are many described in the literature that were intentionally designed for MaOO design problems and would likely have a better convergence success for these types of problems. Integrating one of those into this research's methodology would improve the results even more.

As explained in the literature review portion of this dissertation, the primary limiting reason for not applying other researchers' performance metric categorization methods has been their lack of quantitative support. The research methodology outlined in this dissertation could provide the platform from which these categorization methods could be rigorously tested. This could provide the quantitative evidence needed to justify the use of the categorization methods in performance analysis research.

To facilitate use in other HRT problems and scenarios, this methodology could be better compartmentalized in future work to facilitate distribution and use. Essentially, this would involve turning it into a black box tool with very obvious input variables and structure. This could facilitate a wider range of performance metrics testing, and a wider range of tool application.

A future iteration on the final HRT demonstration problem could experiment with including a larger number of robotic agents in the team configurations. The analysis presented in Chapter 6 limited these teams to two robotic agents. It would be intriguing to measure how teams with multiple independent robotic agents of varying skill capabilities affected the overall performance of the teams.

Future mission designers will have a large heterogeneous group of distinct agents (both human and robotic) from which to select the most productive or efficient team members. The proposed methodology represents a novel solution technique with a wide range of applications. Having a generic, objective methodology like this will be a necessary step to advancing the goal of using cooperative human and robot teams in future space activities.

Appendix A

Previous Cooperative HRT Research

A.1 Preliminary Dissertation Research and Results

Over the last few years, I have done substantial research into characterizing HRT interaction that addressed several of the challenges described in the introduction. That research and its results will be summarized in this section, emphasizing which of the challenges have been researched and which are yet to be addressed.

A.1.1 Methodology to Assess HRT Task Performance

My previous research has sought to characterize how agents in HRTs affect each others' task performance, and how different skill sets influence not only the schedules, but wait time and involvement time for all of the agents. This work led to the development of a methodology [89, 88] to facilitate scheduling a human-robot team according to different mission preferences. The methodology takes into consideration real world constraints and precedent constraints to produce pseudo-optimal schedules for a cooperative human and robotic crew.

The journal paper [90] extended this task allocation and scheduling methodology to assess the impact of the robotic agent's task performance on the cooperative team's schedule. To effectively allocate and schedule mission tasks for a cooperative human and robotic team relies on an accurate characterization of the performance abilities and speeds of the disparate crew.

In terms of addressing the challenges to comparing HRT configurations, this methodology provides a guideline for comparing HRTs before the mission design phase and facilitates quantitative comparison based on crew time as the primary criterion.

A.1.1.1 Case Study: Hubble Space Telescope Servicing Mission 3A

The Hubble Space Telescope Servicing Mission 3A (HST SM-3A) was used as a case study in [89, 88, 90] to demonstrate the scheduling advantages of a cooperative human and robotic team in a space mission application. The flight plans from past Hubble Space Telescope servicing missions (HST SMs) provide a detailed test bed from which to examine the effect of various activities and crew performances in space operations. HST was designed with access panels to allow repair of some of the components of the telescope. Some of the servicing tasks require a fine level of dexterity, including manipulation of tethers and electrical connectors. Most of the tasks can be broken down into primitives that require single degree of freedom motion and utilization of a single tool [68]. The latter set of tasks suggest the improvements and extensions to human performance in space by utilizing robots as cooperative team members who could perform these more repetitive tasks, freeing the humans to work on other tasks.

The six hour nominal EVA day (including daily setup and closeout of HST worksites) is the primary constraint in the number of tasks that can be performed during a servicing mission. Any attempt to reschedule task order is complicated by the need to maintain the sequential constraints between some of the subtasks (e.g., a team member cannot open a door before its securing bolts have been unscrewed) while both minimizing overall human involvement time and minimizing the time one agent has to wait for another.

A.1.1.2 Methodology Assumptions: Relevant Attributes and Criteria

The primary criteria used to assess the value of a cooperative team over the standard two-human crew was the reduction in active human time (representing a specialization of the human crew's activities) and the time that each human crew member needed to wait for the robotic agent to perform its subtasks (a check on the efficiency of the schedule).

The time that the human crew wait for the robot to complete a precedence constrained subtask is the difference between the human involvement time and human active time within a task. This difference represents the time that the human crew was present on-site but not involved in a subtask. It is the human crew's wait time that proved to be the critical factor in determining when it was acceptable to intersperse robot tasks with those of the human crew.

To reduce the dependence of the scheduling model on a specific set of robotic technologies, a flight rule was established for the analysis in [89] that if the robot was active in the same workspace as a human, it would only be doing tasks in support of and that relate directly to the tasks the human was performing (including passing the human hardware), rather than independently performing tasks. A HST flight rule had also been extended to enable the robotic agent to continue servicing activities outside of the human EVA day, when the human crew was no longer on-site. Both of these flight rules continued their relevance to the task performance research and were incorporated into these analyses.

A.1.1.3 Characterizing a Robot's Role in a Servicing Team

Similar to the design of Robonaut, the generic robot used in these analyses was assumed to be capable of configuring its own tools, to have all end effectors necessary to complete the jobs assigned to it, and to be capable of using EVA hardware interfaces (like handholds) and tools. The robot was assumed capable of all force/torque application primitives (including use of EVA tools), hardware handoff and translation, and visual inspection primitives (through either feedback to a controller or autonomous recognition capabilities). It was further assumed to have at least two independently controllable dexterous arms with approximately the same capabilities and reach as a human for sizing around the tasks.

Although the characterization of the robot's mobility and positioning system will be dependent on the specific robot configuration selected, for the purposes of these studies it was assumed that the robot's positioning mechanism would be independent of the human crew's positioning systems. The robot was assumed to be designed with a positioning arm in addition to its assumed dexterous arms, as proposed in [73]. This independence would allow the robot to be added to the human crew's task performance rather than occupying one of the required human positioning aids.

A.1.1.4 Relative Robot Task Performance

The task performance research sought to analyze the role of the robot performance parameter (RPP) in the development of a cooperative team schedule. The RPP represents the relative amount of time that it takes a robot to complete a task with respect to its human analogue. Using the relative performance ratio concept from Rodriguez [74], a value greater than one indicates the robot's performance is slower in magnitude than a human's by the RPP's value. This information can be used to determine a performance bound for the selected robotic agent within its operational capabilities that would reflect the effect that each individual task speed has on the overall team schedule. A robot does not perform every task as quickly as possible, so this level of analysis would seek to match the selected robot's capabilities with the optimal performance required to have it perform and contribute to the development of optimal task performance for the collaborative mission.

Additionally, analysis of the RPP could be used before design selection to influence the rate of task execution, and skill requirements. This would feed preferences derived from an initial scheduling analysis about the role and added benefit of the robotic agent as a crew member on the cooperative team into the design decision.

The RPP analysis examined the extent that precedent constraints between subtasks and primitives create intervals of wait time for the human crew of a cooperative team. These wait time intervals can only be influenced by agent performance speed or reallocation of tasks to different agents. In general, if the wait time is longer than the human performance time of the given subtask, the subtask should be reallocated to the human crew.

A.1.1.5 Role of the Third Agent

It is intuitive that adding a third agent to the servicing team would create schedule benefits by reducing the workload of the original two-human crew through parallel execution of servicing tasks. The role and level of contribution of this third agent depends on the proportion of mission tasks that the third agent has the capabilities to perform, the interdependence of this subset of tasks to those performed by the other crew members, and the rate of task performance and completion of the subset.

In the specific case of an RPP equal to 1, the third agent can be considered as a third human on the crew with restricted task capabilities. To isolate the effect of the RPP on the three agent schedules developed, the flight rules that limited simultaneous task diversity within a cooperative workspace still applied. In particular, the third human was restricted to working on the same task as the other two humans if all three were within the same workspace. A new baseline was established to compare the RPP-varying schedules with a timeline that had been scheduled for a restricted three-human crew.

The RPP research facilitates the ability to more accurately predict how much the human crew's wait time could be reduced by various robotic configurations. Analysis of the effect of the RPP on the cooperative schedule provides framework for using human time window constraints to back out the required robot performance time to maintain an efficient cooperative schedule. This in turn can be used to set robot design requirements.

A.1.2 Results from the HRT Scheduling Research

The efficiency of the human and robot team in the cooperative activities, as defined by Steinfeld et.al. [93], can be measured by the time required for the interaction to be completed. This quantitative metric can be used to compare the contribution of each type of robotic agent to the overall team.

After the subtask-level task allocation analysis, it was noted in [90] that 22% of the HST SM3A mission subtasks needed to be performed by the human crew. For the task allocation strategy specified, this was the minimum solution for EVA time for the mission task profile. The robotic agent defined for this paper was able to aid the human crew in 78% of the mission subtasks. 31% of the mission tasks were reallocated entirely to the robot, freeing the human from involvement in the given task.

One of the most significant results from [89] was that all of the tasks that had been performed during HST SM-3A could be completed in half the active EVA time than during the NASA mission by including the generic robot in the crew. Had a cooperative human and robotic team been employed on the mission as defined in this paper, twice the volume of tasks could have been accomplished during the EVA time of HST SM-3A.

In [88], the impact of three categories of robotic agents, based on their defined role on an operational team, was examined. A result of the analysis was that involving a robotic agent (any of the three categories described) as a third team member to aid in performing the EVA tasks reduced the required human participation in the mission from between 40-60%, with the position in this range depending on the functional capabilities of the robot. With the goal of minimizing human involvement time in the mission while maintaining all constraints, this demonstrates that employing a robotic agent will improve the efficiency of the team over and above the expected contributions of an additional human.

This provides motivation for comparing different HRT configurations and their mission performance possibilities in the early stages of mission design. Had the three-agent crew as defined in [90] been utilized, a much larger quantity of tasks could have been performed during the mission.

A.1.3 Analysis of Previous Scheduling Work

In my previous work, I treated the HRT scheduling problem as an over-constrained, multi-objective optimization problem. The analyses generated significant proof-of-concept results. As a contribution to the field, [89], [88], and [90] present a persuasive argument and supporting evidence that future space operations would benefit from involving cooperative robotic team agents in addition to the human crew. The three papers are based on the same assumptions and the same framing of the HRT problem, allowing direct comparisons between the results in each paper.

These papers define and assess how robots and humans can work together cooperatively to complete tasks, and address several critical issues about their combined performance. A small subset of time-based task attributes were selected as the primary criterion for comparing different team performance. This selection was guided by consideration of team efficiency from a task load model perspective. The methodology guides the application of these metrics through developing a task allocation scheme and a corresponding schedule.

A.1.3.1 Methdology's Quantitative Model

The model described by this methodology incorporated aspects of a task load model. Following a standard task load model, individual agent task completion time was used to gage the effort required for each agent to complete each task. Task times were summed over each EVA day to facilitate using a pairwise comparison of relative time involvement as a common metric between team configurations.

The task allocation scheme used in this methodology was intended to reduce human active time by as much as possible. This metric represented freeing the humans of mundane tasks easily performed by a robot. The new free time would allow humans to specialize their efforts for more complex tasks.

The methodology developed in my previous work, though generic to HRT scheduling problems, represents another scenario specific HRT configuration problem. A human-centered approach was used, such that the primary objective was to reduce active human time during each EVA and a secondary objective was to reduce overall human involvement in each EVA by reordering tasks to group the active intervals. Several other objective functions could have been identified for this analysis - minimize traversal time needed to retrieve tools or visit worksites (application of the traveling salesman problem to increase efficiency of movement), minimize human mental workload, etc. Incorporating all of the objectives that could possibly apply to the HRT scheduling of HST SM-3A would have created an intractable problem.

The objectives used in the analyses were subjectively selected from the larger set of applicable objectives to assess one small portion of the overall HRT problem. This set of reduced objectives does not necessarily fully (if at all) describe other applications or mission scenarios. As discussed by Parasuraman (section 3.7 in [65]), the price of creating quantitative models of the parameters and properties of a HRT is a loss in generality of the resulting model.

The same thing can be said for the constraints used in the analyses. Sequential constraints between tasks were maintained and constraints that restricted the robot's movement in the human workspace were included to preserve a realistic mission scenario. These are also only a small subset of the constraints that could be applied to the larger problem. There are very specific constraints about the amount of time each instrument in the HST can be exposed to space for environmental exposure (limits task time). Power as a resource is not an unlimited quantity - any robot will run off the power of its supporting structure (in this case, the Space Shuttle), and its power draw for the tasks assigned in these analyses could be entirely infeasible.

A.1.4 Challenges Yet to be Addressed

Fully describing every aspect of HRT servicing would yield an intractable problem. To facilitate analysis, assumptions and simplifications must be made to the model to reduce the number of variables. Importance ratings could be applied between the multitude of objective functions. Each of these subtly changes the design space and shapes the resulting solution set, potentially into widely different regions of the design space. Other equally valid selection choices for the important design parameters, objectives, and constraints could have been made. Without using *a priori* information about mission priorities, there are an infinite number of subjective combinations of reduced-complexity problem descriptions.

The question that remains about my previous research is whether there is an objective quantitative method to determine if a neglected parameter would have produced significantly different results. Is it possible to objectively reduce the problem design space to a core of important parameters? With an infinite number of ways to reduce the complexity of a problem, how could a designer objectively determine which is an optimal HRT configuration for any given mission? These are the questions that this dissertation research seeks to answer.

Appendix B

Large HRT MOGA Implementation Details

B.1 Fitness Function Inputs

Several databases were created prior to running the simulations of this experiment. The 40 unique team combinations (defined solely by the task primitives each of the four possible agents were capable of performing) were sent through the task allocation scheme. The result was a task performance schedule for each of the 82 team combinations that specified the order of task performance and which agents were required for each of the subtasks. Each of these team combinations and their corresponding schedules were evaluated across all of the performance metrics specified in section 6.1.3. All of this information (agent task primitive capabilities for each team, team schedules, and team performance data) was stored in a MATLAB data file, to be used as a reference database for the experiment simulation.

It was noted that although there were 82 unique team combinations, there were not 82 unique schedules generated for the database. Due to the simplified task allocation scheme used in this research, several of the teams combinations resulted in the same task allocation and performance schedule. This can be seen in multiple ways. First, if two distinct teams perform the schedule in the same way, then any additional capabilities or skill sets of one of the teams over the other are unused capabilities. The more equipped and advanced robotic agents from these scenarios did not result in schedule or overall team performance improvements. These robotic agents were under-utilized in the schedules. In a further iteration of this research, a function could be written which would add this overlap of schedules from disparate teams into an additional performance metric to gage resource utilization and specialization.

B.2 Fitness Function Outline

The following is a discussion of the functions that were written to evaluate each population generation and return a fitness value for the large HRT application. Each step in the fitness function can be seen in Figure B.1, with the primary details of each step identified. When the MOGA begins, Matlab passes an initial population (the user can specify the size of the population during the MOGA call) of design solutions to the fitness function for evaluation. There were four variables, each bounded to be decimals between zero and one.

The user-defined fitness function initializes by loading the database of input problem information. This includes the 40 unique team combinations, their corresponding task allocation and schedules, and the performance metric data for each of the teams. This information must be provided up front.

The fitness function takes the population generated by Matlab's MOGA and passes it to the function *dv_decimal2int.m*, which converts the design variables to integers. This discretizes the design variables such that the bin size is equal to the inverse of the number of possible agents. There were ten different bins for the large

HRT configuration selection problem: the four agents working independently (agent 1, agent 2, agent 3, and agent 4), and each of the possible pairings of agents required for two-agent tasks (agents 1 and 2, agents 1 and 3, agents 1 and 4, agents 2 and 3, agents 2 and 4, and agents 3 and 4). Each of the design variables was discretized into one of these 10 bins. The output of this function was the list of agents (or pairs of agents) required and specified in each of the design variable combinations of the generation.

It should be noted that the design variables generated for testing by the MOGA were not guaranteed to be feasible solutions. In other words, the MOGA had the ability to generate lists of agents (or empty lists) that were unable to perform the entire task list. Subsequent functions in the fitness function were designed to test for this case and assess the feasibility of each team combination. At this point in the code, however, there was no differentiation.

The function *sch_agentcombos.m* performed the same analysis on the schedules in the database to identify which agents were required for each possible schedule. *Combomatch_kta.m* sought to match the list of agents from the design variables to those required in each of the database's schedules. This function identified the corresponding schedule options for each list of agents in the design variables (i.e. this function takes the list of agents specified by a design point and searches for a task schedule that only requires these agents). A schedule was a match if the exact same list of agents was required in both. A schedule that only required a portion of the design point's list of agents was not a match.

The significance of this schedule match should not be under-estimated. For the

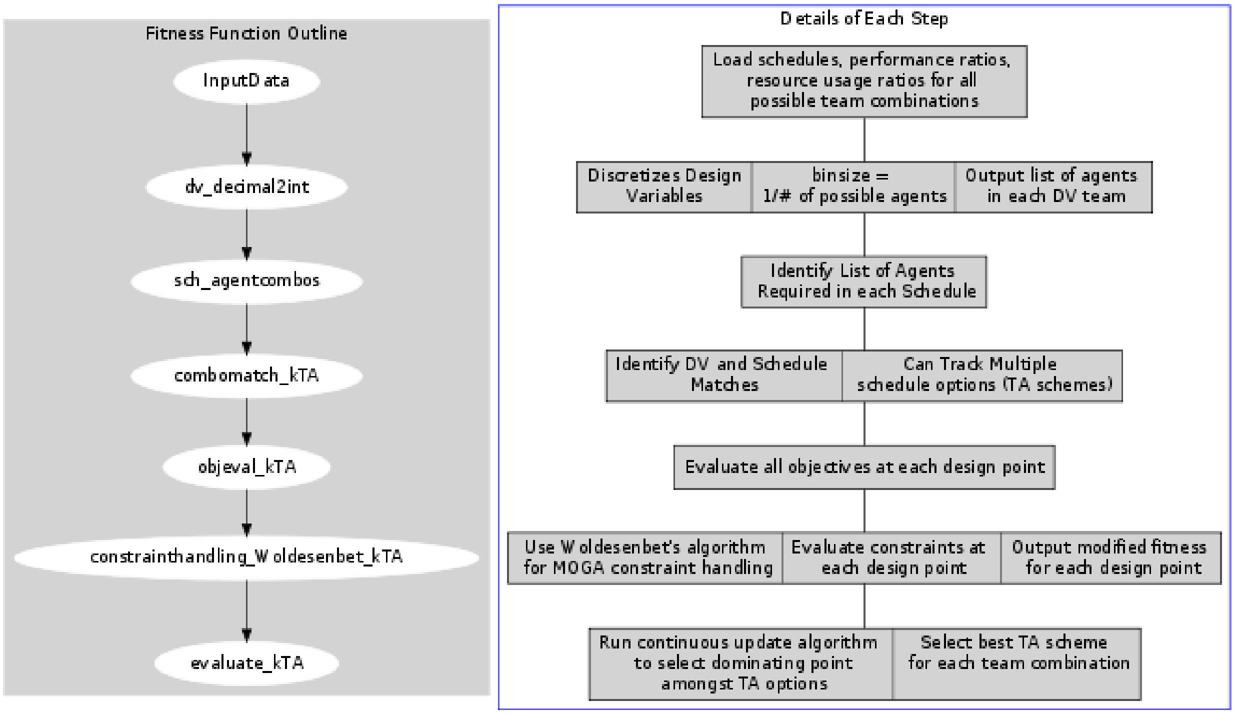

Fitness Function Outline
InputData
dv_decimal2int
sch_agentcombos
combomatch_kTA
objeval_kTA
constrainthandling_Woldesenbet_kTA
evaluate_kTA
Details of Each Step
Load schedules, performance ratios, resource usage ratios for all possible team combinations
Discretizes Design Variables
binsize = 1/# of possible agents
Output list of agents in each DV team
Identify List of Agents Required in each Schedule
Identify DV and Schedule Matches
Can Track Multiple schedule options (TA schemes)
Evaluate all objectives at each design point
Use Woldesenbet's algorithm for MOGA constraint handling
Evaluate constraints at each design point
Output modified fitness for each design point
Run continuous update algorithm to select dominating point amongst TA options
Select best TA scheme for each team combination

Figure B.1: Detailed Diagram of the MOGA Fitness Function Call

given list of participating agents, the schedules represented which agents performed each task. If there were multiple schedule options for a given agent list, then there were multiple task allocation schemes (or multiple options of which agent performs each task) available. At this point in the analysis, it was not possible to select one task allocation schedule from multiple options. Instead, this function and each subsequent function (hence the "kTA" suffix in the function names) kept track of the multiple schedule options for a given design variable. It was only at the end of the fitness function that a final selection was made to determine which task allocation schedule was best for the indicated agent list.

Once the schedules for each design variable were identified, *objeval_kta.m* evaluated each design point for all objective functions. If there was no corresponding schedule for an agent list (no schedules in the database required the specified list of agents), then a filler-value was placed in the performance data for each of the design variables for ease of reference in a later function. This gave the initial evaluation amongst the design points.

The constraints, however, had not been considered, so the feasibility of the solutions was still unknown. A solution that violated constraints was infeasible, and while some of these solutions yielded rewarding objective function values, they were not viable solutions to the problem. The fitness value for each design point must incorporate the feasibility of the point, or the constraints have not been properly applied.

The function *constrainthandling_Woldesenbet_kta.m* performed the feasibility analysis on each of the design points. The multi-objective, adaptive constraint handling technique described by Woldesenbet [101] was implemented in this function (described in more detail in section 3.2. All constraints were coded into this function. The function evaluated the constraints at each of the design points, and then used Woldesenbet's algorithm for MOGA constraint integration to create a modified fitness value for each design point. This modified fitness value was calculated from the objective function values, with a penalty applied for each violated constraint. The weighting of the importance of constraint violation in the fitness evaluation depended on the number of feasible solutions in each generation. In generations with few feasible solutions, violated constraints represented a smaller penalty to enable information from these solutions to be fed into the diversity of a subsequent generation.

The last function called by the fitness function determined which schedule to use in evaluating a design point if there were multiple available (multiple task allocation options available). It called the continuous update domination algorithm (discussed in section 3.2) to select the dominating point between the schedule options. The best schedule's fitness value was then selected to be the modified fitness for the design point. All of the modified fitness values from the constraint handling algorithm were returned to the MOGA to be used in selecting the next generation of design points.

B.3 Fitness Function Output

In practice, it was discovered that several of the schedule options for a given agent list represented non-dominated solutions. Although they corresponded to different objective function values for each of the performance metrics, these solutions did not dominate each other (as determined by the continuous update domination algorithm). To reflect this, a modification was made to the *evaluate_kta.m* function to return the list of teams that represent equally valid and equivalently performing solutions. This information was not utilized within the MOGA itself, but was used in post-simulation processing. For the MOGA itself, one of the feasible, equally-valid solutions was selected for the modified fitness values to enable the MOGA to continue churning a new generation.

B.4 Stochastic MOGA Usage

When running the MOGA for this set of simulations, the built-in MOGA control options were altered. The MOGA options used previously had been intended to run through 100 generations and then stop the search process. The changes made for this experiment was to have the MOGA run through its entire search process and come up with its best Pareto front for the given problem. The MOGA was set to run for 400 generations in each of the problem instances, maintaining a population of 200 individuals, until either the maximum generation counter was reached or the average spread in the Pareto front was on the order of 1e-06 (tolerance to determine that the Pareto front has converged).

A single run through a genetic algorithm might not have searched the entire design space. It is possible that the algorithm could become stuck in a local minima, resulting in non-reproducible results and sub-optimal solutions. To avoid this type of narrow-search complication, a stochastic run wrapper was designed to run the genetic algorithm multiple times, keeping track of the non-dominated solutions from all of the runs (see Figure 3.5 and section 3.2.2). The wrapper function has the ability to call the multi-objective algorithm any number of user-specified times (10 is recommended as a general rule-of-thumb for statistical relevance). This stochastic solution set better describes the design space. Over the iterative runs, a larger swath of the design space will be searched, and the algorithm has a better chance of converging on a steady state solution set.

As discussed in section 3.2.2, an algorithm is called from the stochastic run wrapper to assess if any of the solutions from each run of the MOGA dominate each other. The non-domination algorithm implemented is the continuous update method from Deb [22], but has been expanded in this research to accommodate multiple objectives during the domination evaluation. The continuous update method is a faster computational method that does not check all solutions before deciding domination. Instead, it keeps a running list of non-dominated solutions. When an item in this list is dominated, it is removed from the list. When a new design point is demonstrated to be non-dominated by any other solution in the data set, then that point is added to the non-dominated list. After each stochastic run, the new Pareto set is compared with the existing non-dominated set, and the overall set of non-dominated points results.

The final set of non-dominated points across all of the stochastic runs represented the final Pareto front solution to the large HRT configuration selection problem. To identify which of the 82 unique teams were best suited to the mission, a final run through the fitness function was conducted with the final set of non-dominated points. The design variables were discretized to reflect the agent list. The uniqueness of these lists was then sought. Unsurprisingly, there was substantial overlap amongst the discretized agent lists. A unique subset was down-selected, and it was these design points that were funneled through the rest of the analysis. Each was assessed for schedules and performance metric values.

Rather than using the final modified fitness value for each of the design points, however, the *evaluate_kta.m* function was used to return the list of each equivalently performing team for each of the design points. The final return from this function was a list of the best equivalently performing teams from the 40 unique possible teams.

Appendix C

Large HRT Configuration Selection Problem - Initial Details and Results

An initial instantiation of the large-scale HRT configuration selection problem was developed and run prior to the experiment described in Chapter 6. The Chapter 6 experiment was designed to yield more meaningful results than the first run of the experiment had. The details of that experiment and its results are contained in this Appendix chapter. Only those experimental details that were different from those described in Chapter 6 are herein related for brevity.

C.1 Initial Setup: Hubble Space Telescope Servicing Mission 3A

For this experiment (as with the experiment discussed in Chapter 6), the mission profiles from the Hubble Space Telescope Servicing Mission 3A were selected to form the background of this performance analysis. HST Servicing Mission 3A (HST SM-3A) provides a useful platform to analyze role definition of each of the mission agents, influences of task allocation schemes, the resulting cooperative schedules, and the overall performance of the combined human and robot servicing team.

The flight plans from past Hubble Space Telescope servicing missions (HST SMs) (as represented in the HST SM-3A EVA Checklists [100]) provide a detailed data set from which to examine the effect of various activities and crew performances on space operations. HST was designed with access panels to allow repair of the components of the telescope. Some of the servicing tasks required a fine level of dexterity, including manipulation of tethers and electrical connectors.

C.1.1 Task Primitive Data Preparation

The actual rate of task performance for a specified robotic agent will vary for different categories of tasks, and between iterations of the same task. As with humans, a robot might not perform a task at the exact same rate on two different attempts. There will be competing priorities in task performance including power and energy resource utilization, measurement errors and overshoot correction, differences in environmental factors including lighting, situational awareness, and maintaining safe operating conditions in a crowded workspace. All of these factors will affect the robot's actual task performance speed and will vary between tasks.

The subtask completion time data used in this analysis is identical to that originally anticipated from the HST SM-3A mission. Rather than add uncertainty by modifying this time data to define whether robotic agents perform subtasks faster or slower than a human, this *same time data* was used for the robot subtask completion times in this experiment.

The robot time data values used in this analysis provide an initial guide to enable scheduling analysis for a cooperative human and robotic team. These values could be updated to reflect a specific robotic system design if desired by future users.

C.1.2 Team Characterization and Scheduling

Forty unique teams were generated for this experiment. Each team had at most four agents, two human and two robotic. Each of the agents in these teams were defined based on which of the nine task primitive types they were capable of performing. The teams were specified with a matrix for each, with the rows corresponding to the four possible agents, and each column representing a task primitive type. A value of one in the matrix meant that the specified agent could perform the task primitive type. A value of zero meant that the agent did not have the skill set or capacity to perform the task primitive type.

C.1.3 Performance Metrics

The same performance metrics were used for this experiment and for the experiment described in Chapter 6. However, the rating values used in this version of the experiment were solid integers and the 20% random variable was not included. These alterations affected the performance metric data for the following metrics:

- Human inactive time
- Total human involvement time
- Total mission duration time
- Total robot involvement time
- Robot inactive time
- Human physical workload

- Human mental workload
- Autonomous navigation speed
- Approachability
- Autonomous traverse distance

In addition to the alterations listed above, the expected value rating scale on the three survey performance metrics was increased for the experiment described in Chapter 6. For this first version of the experiment, the autonomous navigation speed rating scale was as follows:

- If both of the potential robotic agents in a team pair were capable of traversal, the autonomous navigation speed rating of the team pair was given a value of 1.
- If either of the two potential robotic agents in a team pair were capable of traversal, the autonomous navigation speed rating of the team pair was given a value of 5.
- If either of the two potential robotic agents in a team were present and active (if either was capable of at least 1 task primitive type) but both were incapable of independent movement, the autonomous navigation speed was given a rating value of 15.
- If there were no robots in a given team combination, then the human pair were given a rating value of 1, equivalent to the fully functional robotic reference system.

Similarly, the autonomous traverse distance performance metric had a similar valued rating scale, such that:

- The reference system for this performance metric was two fully functioning robotic agents, capable of both traversal and visual inspection. A rating value of 1 was given to this type of system.
- If only one robot was capable of traversal and visual inspection, this robot earned a rating of 1, equivalent to the fully functioning candidate system.
- If only one robot was capable of traversal and visual inspection but both were capable of traversal, a rating value of 5 was earned to denote the added workload required by the one robot to provide visualization for both robotic agents.
- If only one robot was capable of traversal, but this robot was not capable of visual inspection, the team earned a rating value of 5.
- If either robot was present and active (if either was capable of at least one task primitive type) but both were incapable of traversal and visual inspection, the team earned a rating value of 15.
- As with the previous metric, if there were no robots active for a candidate team (i.e. a human-only team), an autonomous traverse distance rating value of 1 was used.

The approachability performance metric was scaled in the same fashion:

- If a robotic system was capable of each of these three task primitive types, it earned an approachability rating value of 1.

- If a robotic system was capable of traversal and visual inspection but did not carry an on board tool, it earned a rating value of 3. While this might seem a generous rating, it was anticipated that the robotic agent would find and locate targets. A human agent would then arrive at the specified target and use the instrument.

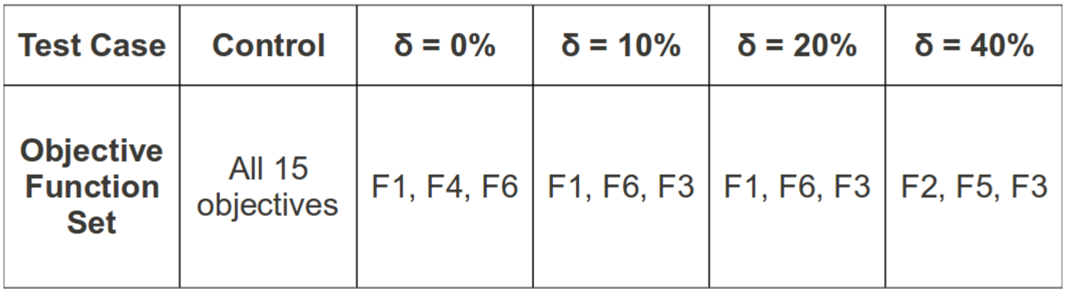
- If neither robot was capable of traversal, visual inspection, or carrying a tool, an approachability rating value of 10 was used.

- As with the previous metric, if there were no robots active for a candidate team, an approachability rating value of 1 was used indicating equivalent performance to the robotic reference system.

C.2 Results

This experiment followed the analysis steps laid out in the methodology discussed in Chapter 3. The reduced objective sets were used to run the MOGA 10 times for statistical relevance. Each run sought to reach convergence of the Pareto set. The output from all of the runs were collated and sent a final iteration through the continuous update algorithm to verify that they represented a set of nondominated solutions. The results from this final analysis can be seen in Figure C.2.

Figure C.2 displays each of the 40 unique teams that were feasible solutions for



Test Case	Control	δ = 0%	δ = 10%	δ = 20%	δ = 40%
Objective Function Set	All 15 objectives	F1, F4, F6	F1, F6, F3	F1, F6, F3	F2, F5, F3

Figure C.1: Reduced Objective Sets for the First Instantiation of the Large HRT Demonstration Problem in Each of Five Test Cases

the large HRT demonstration problem on the x-axis. On the y-axis are the five test cases that were run (vertically offset for visibility only - there is no significance to height on the y-axis). From Figure C.2 it is apparent that the Pareto sets of solutions resulting from the δ-MOSS analysis not only resulted in good approximations of the control Pareto set (resulting from all 15 objectives), but in several of the test cases, they were the exact same results.

This result is the final justification that was sought from running the large HRT demonstration problem - application of the methodology proposed in this dissertation provided the rigorous performance analysis that had been missing from previous research in the literature. The δ-MOSS algorithm provided objective, analytical reasoning for selecting some performance metrics and deeming others redundant. Rather than running a comprehensive performance analysis with 15 objectives, a researcher could objectively and conclusively run the performance analysis with 3 objectives at any of the given error-tolerances and achieve the same results.

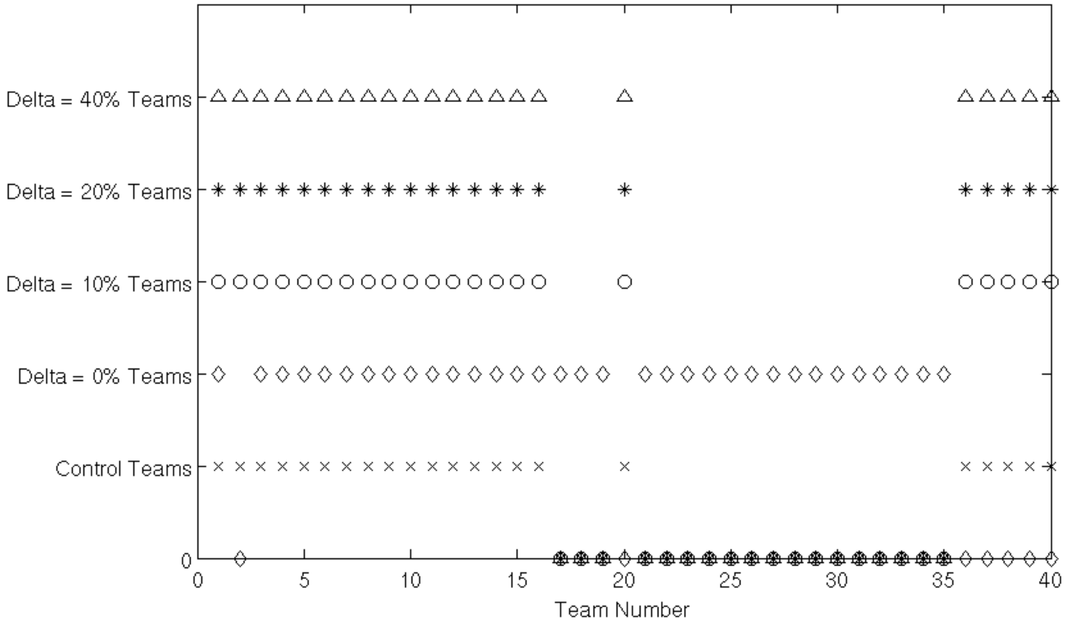


Figure C.2: Teams Selected for Large HRT Demonstration Problem in Each of Five Test Cases

Figure C.3 demonstrates the convergence success for each of the test problems across each of the 10 stochastic runs through the MOGA. The control case (with 15 objectives) converged in only 7 out of 10 of the runs. This means that 30% of the runs failed to find the Pareto front - the MOGA had too many competing objectives. This was approximately the convergence rate of the MOGA during the 15 objective knapsack test problem instance.

The convergence results from the $\delta = 0\%$ test case were more concerning. Only 30% of the runs reached convergence. This suggests that the team analysis of this test case should be taken with a grain of salt - the results from this test case do not represent a final Pareto front. Even with only 3 objectives, the MOGA failed to converge an unacceptable percentage of the time. However, the opposite results

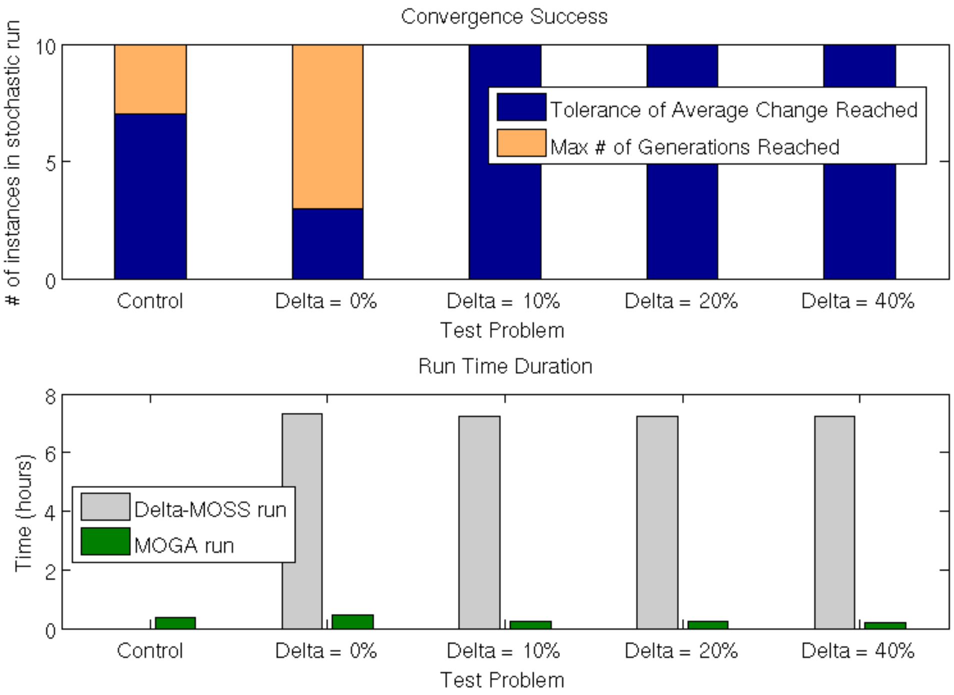


Figure C.3: Termination Reason and Run Time for Large HRT Demonstration Problem

were seen in the $\delta = 10\%$, 20%, and 40% test cases. Each represents three objective functions run to complete convergence within the MOGA. All three test cases had 100% convergence rate.

A combination of the results from Figure C.3 and the results of Figure C.1 yield the result that the reduced performance metric set for this instantiation of the large HRT demonstration problem should contain either total mission performance time, human inactive time, and human physical workload, or human physical workload, total EVA involvement time (length of the EVA day), and total robot involvement time. Either of these two performance metric sets resulted in the same Pareto set of solutions and both resulted in 100% convergence of the MOGA.

As had been the case with the knapsack test problem instances from Chapter 4, the computational time required for the δ-MOSS algorithm was the only disadvantage of this method. As seen in Figure C.3, the δ-MOSS algorithm required more than 7 hours to run for each of the test problems. In comparison, the control test problem's MOGA required less than an hour to complete. As had been concluded from the knapsack test problems, if computational time is a limiting factor for future performance analysis, this methodology would not be advantageous.

However, if computational time is not an issue for *a priori* analysis, a significant reduction in problem complexity can be achieved by utilizing the methodology proposed in this dissertation. A final analysis on how good the Pareto sets resulting from each of these five test cases was performed, utilizing the hypervolume quality metric (see section 3.2.4). These results are tabulated in Figure C.4.

There are several important trends to note from the data represented in Figure

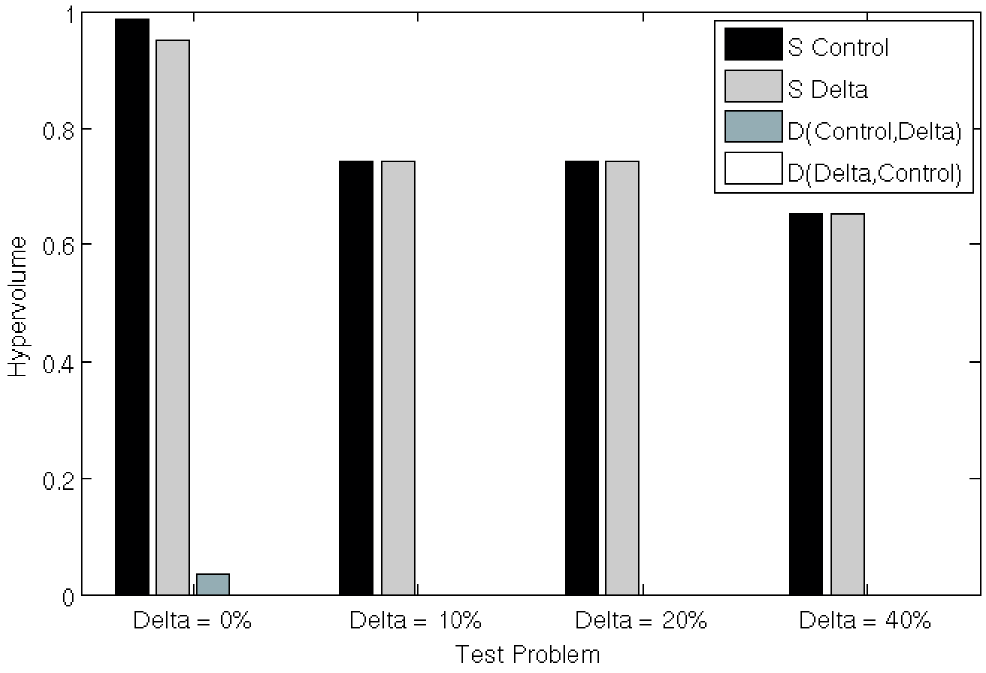


Figure C.4: Hypervolume Indicator Quality Metric for Each Test Case of Large HRT Demonstration Problem

C.4. Recalling that the S quality metric reflects absolute volume coverage of the Pareto front, it can be seen for each of the four test problems that the reduced objective set resulted in a Pareto front with nearly identical volume coverage. Each Pareto front generated from this analysis represented remarkably good coverage across all of the objectives.

Recall that the D quality metric was the relative volume coverage used to compare two different Pareto sets. In this large HRT problem, the control was compared across each of the four test cases with different δ-error values. In the δ = 0% test case, the Pareto volume coverage of the control set was approximately 5% better than the Pareto volume coverage resulting from the reduced objective set. This result is consistent with those seen in section 3.2.4 for the knapsack test problem.

However, the results from the other three test problems in the large HRT problem are impressive. An exact value of zero was calculated for the relative volume coverage both comparing control to the δ-Pareto set, and δ-Pareto compared to the control. There were no portions of the Pareto front that were covered by one set and not by the other. This provides the analytical support for the assertion that the Pareto sets resulting from the reduced objective sets not only provide a good approximation of the control Pareto set (in general), but (in this case) provided *the exact same coverage of the Pareto front.* Zero error would be introduced if a mission designer chose to use the Pareto set generated by the 3-objective test cases instead of the 15-objective control case.

C.2.1 Analysis of Bias in Solution Selection

Application of this research's methodology greatly reduced the large HRT configuration selection problem from a 15-objective across 40 teams decision down to a 3-objective across 22 teams decision. Several questions arose from the previous analysis. Were the 22 solution teams (winners) that made the final cut uniquely qualified and higher performing than the other 18 teams (losers) or was there some level of biasing in the final solution set due to a majority vote type of situation?

To further examine the relations between the solutions, an additional run through the methodology was arranged for the 18 teams that were pruned out of the final solution set. One team from the winning set was added to be representative of the winning solutions. The large HRT configuration selection problem was run again with only these 19 candidate teams as solution options across the full 15 objective functions.

The first step in the methodology process, however, yielded unexpected results. An initial run through the MOGA for 100 generations had been expected to provide an initial population of good-ranking teams to input into the δ-MOSS objective reduction algorithm. However, after 100 generations (iterating through the 15 objective functions and 19 candidate solutions) the MOGA returned a singular best team result. The MOGA identified the 1 winner candidate solution as the best overall from the 19 candidate solutions.

Several different winner candidate solutions were used in the 19 candidate solutions (only 1 per iteration) to determine if the MOGA would be enticed to select one of the other 18 solutions. On every iteration, however, the one winner team configuration was selected, regardless of which of the winner teams was used.

This result reinforces the reproducibility of the final configuration selection results. When all chances of majority vote team biasing had been removed from the candidate teams, the one winner candidate solution stood out as the best overall configuration for the given HRT configuration selection problem. The MOGA analysis was run 10 times to verify this result. Every stochastic iteration returned that one candidate winner solution as the best solution.

Even though a best team had been found for this additional analysis, it was still desirable to assess the objective reduction results from this smaller loser-team population. The initial 100 generation MOGA population contained only versions of that one winner candidate solution, so the objective function values themselves for the 19 candidate teams were used as input to the δ-MOSS algorithm. The objective values were scaled (as had been done in previous iterations of the δ-MOSS algorithm) such that the difference between the largest and smallest objective function value would represent a δ-error equal to 1.

The results from the objective reduction algorithm were intriguing. With zero δ-error allowed in the underlying dominance structure, the 15 performance metrics were reduced down to 5 ($F\prime = \{F_2, F_{13}, F_1, F_{12}, F_{15}\}$), ordered by importance to the underlying dominance structure) to fully represent these 19 candidate teams. When a 5% δ-error was tolerated, the reduced performance metric set was reduced to only 2 objectives ($F\prime = \{F_1, F_3\}$). As the δ-error value was increased, this reduced objective set held constant and was unchanged both in terms of objective ordering and makeup of the reduced set. A δ-error value of 35.5% reduced the objective set to a single performance metric ($F\prime = \{F_1\}$). All higher error tolerance levels maintained this one performance metric in the reduced set.

Without the winner candidate teams in the set, the objective reduction algorithm was only able to reduce the objective set size to 5 objectives rather than the 3 it had found in the full analysis. Referring back to Figure C.1, it is interesting to note that three of the objectives chosen in this last δ-MOSS algorithm iteration were never selected by the algorithm in the full set. The objectives that differentiated the loser candidate solutions from each other were irrelevant for the winner candidate teams. Part of the reason for this was that the loser candidate teams had similar performance across many of the objectives, to an extent that those objectives were no longer valuable in differentiating the solutions.

Including a δ-error of 5%, however, reduced the objective set to $F\prime = \{F_1, F_3\}$. Both of these objectives were deemed highly valuable to defining the underlying dominance structure of the original large HRT configuration selection problem. Both of these objectives turned up in several of the δ-error cases depicted in Figure C.1. Metric $\{F_1\}$ reflects a preference for minimizing total mission duration time, and $\{F_3\}$ sought to minimize overall human physical workload. It can be inferred that the primary differences between the loser candidate teams reflected the usage of the human crew as a resource on the team. Each of these teams had at least one robotic crew member, but the robotic crew were all tasked in the same way.

A final question that should be asked in this analysis is what this variation in the reduced objective set components say about the reproducibility of the reduced objective set for selecting performance metrics. In the case where two performance metrics contain equivalent problem data and represent the same pairwise dominance relations, the first encountered in the δ-MOSS iterations will be selected for placement into the reduced objective set. In the next iteration, therefore, all of the relations contained in the second equivalent performance metric will already be represented in the basis. This second performance metric will not be selected to be included in the basis. While the initial input ordering of performance metrics will obviously have an affect on those selected for the reduced set, the pairwise dominance relations represented by the final reduced set will remain the same. In other words, a variation in the performance metrics used in an overall analysis, if varied by the δ-MOSS algorithm, will not have an affect on the resulting Pareto solution sets.

In a sense, all of the problem detail can be represented by these candidate solutions. The algorithm determines which objective functions provide the most information for the given candidate solutions. When a set of poor candidate solutions were input into the δ-MOSS algorithm, a different set of objective functions was deemed necessary to reflect the underlying dominance structure than when a set of mixed good and poor candidate solutions were input into the algorithm.

From this perspective, it would be expected that the quality of the solutions would cause variation in the reduced objective set. When applying the δ-MOSS algorithm, the reduced set of objectives will be selected based on non-arbitrary, problem-dependent information. No preference information will be needed. A large number of candidate objectives can be used to canvas the decision space, or a smaller set could be used. Either way, the algorithm will isolate the underlying dominance structure of the problem domain and identify those performance metrics that are most needed to characterize the decision space. The objective reduction algorithm has proved itself an immensely valuable tool for identifying the critical information for various problem types and across different domains.

C.2.2 Pareto Decision Space Analysis

It has been demonstrated that the Pareto solution sets generated by utilizing this dissertation's proposed methodology are approximately (if not exactly) equivalent to using a large, unreduced objective set. It has been analytically demonstrated that large, complex, multi-objective optimization problems can be generically, objectively, and conclusively reduced by application of this methodology.

How much does this methodology aid a mission designer's decision making? There has been significant reduction in the complexity of the mission designer's decision space. Where previously the mission designer had a design space detailed by 40 unique teams compared across 15 performance metrics, this methodology reduces the design space to 22 unique teams compared across 3 performance metrics. This is a substantial improvement for the mission designer.

It could be argued that 22 unique teams is still too many for a mission designer to need to compare simultaneously. This number, however, reflects shortcomings of the input data rather than the method itself. Although there were 40 unique teams input into the beginning of this analysis, it was noted that this did not result in 40 unique team schedules. The task allocation schema used in this analysis to generate the schedules was very simplistic, resulting in several of the unique teams generating identical schedules.

A further iteration on this experiment could apply a more complex and selective task allocation schema that would further differentiate the schedules and, therefore, the 40 team options. It would be anticipated that this would further reduce the number of candidate team options represented by the Pareto front, and further reduce the options that a mission designer must choose from.

This experiment was able to significantly reduced the designer's decision space. Additionally, the designer was analytically assured that the results from this reduced decision space were at least as good as a solution selected from the design space represented by the control's solution set.

C.3 Summary

This experiment sought to demonstrate the methodology proposed by this dissertation on a large-scale HRT configuration selection problem. The other experiments from this dissertation had suggested that the application of the methodology to this design space would facilitate significant reduction in the complexity of the over-constrained, multi-objective optimization problem, and that it would provide the rigorous objective reasoning to down-select from a large set of performance metrics to the few significant ones for analysis.

40 unique teams were proposed as candidate solutions and were compared across 15 performance metrics. For a mission designer, this is a very large decision space to consider. In the research described in Chapter 2, previous mission designers would have arbitrarily selected a small set of performance metrics to use in an overall team performance analysis. The lack of rigor in this type of analysis meant that comparison of results between different researchers, platforms, and team pairings was virtually impossible.

Applying the δ-MOSS objective reduction algorithm to this problem immediately (and rigorously) reduced the performance metric set from 15 down to 3 performance metrics. Although there was some variability in the reduced set depending on the tolerable error in the underlying problem structure, this significantly reduced the decision maker's design space. This by itself was a significant result.

This research's methodology, however, took the analysis one step further to assess the Pareto solution sets that resulted from the reduced objective sets. It was this final analysis step that reduced the 40 unique teams down to the 22 best teams.

The methodology proposed in this dissertation research has been applied to three different application realms and has proved beneficial in all three. This methodology has wide utility in reducing the complexity of large-scale over-constrained, multi-objective optimization problems. It would be anticipated that the methodology will provide rigorous analysis on many future applications.

Bibliography

[1] H. Aguirre and K. Tanaka. Adaptive epsilon ranking on many objective problems. *Evol. Intel.*, 2(4):183–206, 2009.

[2] D.L. Akin, S. Jacobs, and D. Gruntz. Investigations into several approaches to eva-robot integration. In *37th International Conference on Environmental Systems*, number SAE 2007-01-3232. SAE International, 2007.

[3] D.L. Akin, B. Roberts, K. Pilotte, and M. Baker. Robotic augmentation of eva for hubble space telescope servicing. In *AIAA Space Conference Exposition*, number AIAA-2003-6274. AIAA, 2003.

[4] J.A. Arnold. Towards a framework for architecting heterogeneous teams of humans and robots for space exploration. Master's thesis, Massachusetts Institute of Technology, 2006.

[5] J. Bader and E. Zitzler. A hypervolume-based optimizer for high-dimensional objective spaces. *New Developments in Multiple Objective and Goal Programming*, pages 35–54, 2010.

[6] A. Bechar, Y. Edan, and J. Meyer. Optimal collaboration in human-robot target recognition systems. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 5, pages 4243–4248. IEEE, 2007.

[7] A. Bechar, J. Meyer, and Y. Edan. An objective function to evaluate performance of human-robot collaboration in target recognition tasks. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(6):611–620, 2009.

[8] L. Billman and M. Steinberg. Human system performance metrics for evaluation of mixed-initiative heterogeneous autonomous systems. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, pages 120–126. ACM, August 2007.

[9] P. Bobko, P. Roth, and M. Buster. the usefulness of unit weights in creating composite scores: a literature review, application to content validity, and meta-analysis. *Organizational Research Methods*, 10(4):689–709, 2007.

[10] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 765–772. ACM, 2007.

[11] D. Brockhoff, D.K. Saxena, K. Deb, and E. Zitzler. On handling a large number of objectives a posteriori and during optimization. *Multiobjective Problem Solving from Nature*, pages 377–403, 2008.

[12] D. Brockhoff and E. Zitzler. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation*, 17(2):135–166, 2009.

[13] D.J. Bruemmer, D.A. Few, R.L. Boring, J.L. Marble, M.C. Walton, and C.W. Nielsen. Shared understanding for collaborative control. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4):494–504, July 2005.

[14] J.L. Burke, R.R. Murphy, D.R. Riddle, and T. Fincannon. Task performance metrics in human-robot interaction: Taking a systems approach. *Performance Metrics for Intelligent Systems*, 2004.

[15] J.W. Crandall and M.L. Cummings. Developing performance metrics for the supervisory control of multiple robots. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40. ACM, 2007.

[16] J.W. Crandall and ML Cummings. A predictive model for human-unmanned vehicles systems. Technical Report HAL2008-05, Massachusetts Institute of Technology, 2008.

[17] J.W. Crandall, M.A. Goodrich, D.R. Olsen, and C.W. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4):438–449, July 2005.

[18] J.W. Crandall, C.W. Nielsen, and M.A. Goodrich. Towards predicting robot team performance. In *Systems, Man and Cybernetics, 2003 IEEE International Conference on*, volume 1, pages 906–911. IEEE, 2003.

[19] ML Cummings, P. Pina, and J.W. Crandall. A metric taxonomy for supervisory control of unmanned vehicles. In *AUVSI*, 2008.

[20] K. Dautenhahn and I. Werry. A quantitative technique for analysing robot-human interactions. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[21] E. De Visser, R. Parasuraman, A. Freedy, E. Freedy, and G. Weltman. A comprehensive methodology for assessing human-robot team performance for use in training and simulation. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, volume 50, pages 2639–2643. Human Factors and Ergonomics Society, 2006.

[22] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley and Sons, 2001.

[23] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms.* Wiley, 2008.

[24] B. Donmez, P. Pina, and M.L. Cummings. Evaluation criteria for human-automation performance metrics. In *Proceedings of the Performance Metrics for Intelligent Systems Workshop*, 2008.

[25] M.R. Elara, C.A.A. Calderon, C. Zhou, and W.S. Wijesoma. False alarm demand: A new metric for measuring robot performance in human robot teams. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pages 436–441. IEEE, 2009.

[26] A. Elfes, C.R. Weisbin, H. Hua, J.H. Smith, J. Mrozinski, and K. Shelton. The huron task allocation and scheduling system: Planning human and robot activities for lunar missions. In *Automation Congress, 2008. WAC 2008. World*, pages 1–8. IEEE, 2008.

[27] C.A. Ellis, S.J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 34(1):39–58, 1991.

[28] M. Farina and P. Amato. A fuzzy definition of optimality for many-criteria optimization problems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(3):315–326, 2004.

[29] T. Fong, A. Abercromby, M.G. Bualat, M.C. Deans, K.V. Hodges, J.M. Hurtado, R. Landis, P. Lee, and D. Schreckenghost. Assessment of robotic recon for human exploration on the moon. *Acta Astronautica*, 67(9-10):1176–1188, November–December 2010.

[30] T. Fong, M. Bualat, M. Deans, M. Allan, X. Bouyssounouse, M. Broxton, L. Edwards, R. Elphic, L. Flückiger, J. Frank, et al. Field testing of utility robots for lunar surface operations. In *AIAA Space*. Citeseer, 2008.

[31] T. Fong, I. Nourbakhsh, C. Kunz, L. Fluckiger, J. Schreiner, R. Ambrose, R. Burridge, R. Simmons, L.M. Hiatt, A. Schultz, et al. The peer-to-peer human-robot interaction project. *Space*, 6750(AIAA 2005-6750), September 2005.

[32] T. Fong, J. Scholtz, J.A. Shah, L. Fluckiger, C. Kunz, D. Lees, J. Schreiner, M. Siegel, L.M. Hiatt, and I. Nourbakhsh. A preliminary study of peer-to-peer human-robot interaction. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 4, pages 3198–3203. IEEE, 2007.

[33] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman. Measurement of trust in human-robot collaboration. In *Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on*, pages 106–114. IEEE, 2008.

[34] B.P. Gerkey and M.J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[35] D.F. Glas, T. Kanda, H. Ishiguro, and N. Hagita. Simultaneous teleoperation of multiple social robots. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, pages 311–318. ACM, 2008.

[36] J. Glassmire, M. O'Malley, W. Bluethmann, and R. Ambrose. Cooperative manipulation between humans and teleoperated agents. In *Proceedings of the IEEE International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004.

[37] M. Goodrich and D. Olsen. Seven principles of efficient human robot interaction. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 3943–3948, 2003.

[38] M.A. Goodrich, T.W. McLain, J.D. Anderson, J. Sun, and J.W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 25–32. ACM, March 2007.

[39] S.G. Hart and L.E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human Mental Workload*, 1:139–183, 1988.

[40] F. Heger and S. Singh. Sliding autonomy for complex coordinated multi-robot tasks: analysis and experiments. In *Proceedings of Robotics: Science and Systems*, August 2006.

[41] A. Howard. A synergistic approach for maximizing human-automation system performance. Draper fiscal year 2006 final report, Georgia Institute of Technology, July 2006.

[42] A. Howard. A systematic approach to predict performance of human-automation systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 37(4):594–601, July 2007.

[43] A. Howard and G. Cruz. Adapting human leadership approaches for role allocation in human-robot navigation scenarios. In *Automation Congress, 2006. WAC'06. World*, pages 1–8. IEEE, 2007.

[44] E.J. Hughes. Fitness assignment methods for many-objective problems. *Multiobjective Optimization*, pages 307–329, 2008.

[45] J.H. Hwang, K.W. Lee, and D.S. Kwon. A formal method of measuring interactivity in hri. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 738–743. IEEE, 2007.

[46] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. IEEE World Congress on Computational Intelligence*, pages 2419–2426. IEEE, 2008.

[47] A.L. Jaimes, C.A.C. Coello, H. Aguirre, and K. Tanaka. Adaptive objective space partitioning using conflict information for many-objective optimization. *Evolutionary Multi-Criterion Optimization*, 6576:151–165, 2011.

[48] B. Kannan and L.E. Parker. Fault-tolerance based metrics for evaluating system performance in multi-robot teams. In *Proceedings of Performance Metrics for Intelligent Systems Workshop*, 2006.

[49] T. Kaupp and A. Makarenko. Measuring human-robot team effectiveness to determine an appropriate autonomy level. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2146–2151. IEEE, 2008.

[50] T. Kaupp, A. Makarenko, and H. Durrant-Whyte. Human-robot communication for collaborative decision making - a probabilistic approach. *Robotics and Autonomous Systems*, 58:444–456, 2010.

[51] J. Keller. Human performance modeling for discrete-event simulation: human performance modeling for discrete-event simulation: workload. In *Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers*, pages 157–162. Winter Simulation Conference, 2002.

[52] C. R. Kurtzman. *Time and resource constrained scheduling, with applications to space station planning.* PhD thesis, Massachusetts Institute of Technology, 1988.

[53] A. Lampe and R. Chatila. *Performance measure for the evaluation of mobile robot autonomy.* Proceedings of the 2006 IEEE International Conference on Robotics and Automation, May 2006.

[54] A. López Jaimes, C. Coello, and J. Urías Barrientos. Online objective reduction to deal with many-objective problems. In *Evolutionary Multi-Criterion Optimization*, pages 423–437. Springer, 2009.

[55] I.S. MacKenzie. A note on the information-theoretic basis for fitts' law. *Journal of Motor Behavior*, 1989.

[56] G.A. Mann. Quantitative evaluation of human-robot options for maintenance tasks during analogue surface operations. In C. Pain, editor, *Proceedings of the 8th Australian Mars Exploration Conference*, pages 26–34, 2008.

[57] J.L. Marble, D.J. Bruemmer, D.A. Few, and D.D. Dudenhoeffer. Evaluation of supervisory vs. peer-peer interaction with human-robot teams. In *Proceedings of the 37th Annual Hawaii International Conference on Systems Sciences*, pages 4–8, 2004.

[58] C.A. Miller and R. Parasuraman. Who's in charge?: Intermediate levels of control for robots we can live with. *Systems, Man, and Cybernetics, 2003. IEEE International Conference on*, 1:462–467, 2003.

[59] C. E. Nehme. *Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems.* PhD thesis, Massachusetts Institute of Technology, February 2009.

[60] U. Nehmzow. Quantitative analysis of robot-environment interaction–towards scientific mobile robotics. *Robotics and Autonomous Systems*, 44:55–68, 2003.

[61] D.R. Olsen and M.A. Goodrich. Metrics for evaluating human-robot interactions. In *Proceedings of PERMIS*, 2003.

[62] D.R. Olsen, B. Wood, and J. Turner. Metrics for human driving of multiple robots. In *International Conference on Robotics and Automation*. IEEE, 2004.

[63] D.R. Olsen and S.B. Wood. Fan-out: measuring human control of multiple robots. In *Proceedings of the Special Interest Group on Computer Human Interaction Conference on Human Factors in Computing Systems*, pages 231–238. ACM, 2004.

[64] Y. Oren. Performance analysis of human-robot cooperation in target recognition tasks. Master's thesis, Ben-Gurion University of the Negev, Aug. 2008.

[65] R. Parasuraman. Designing automation for human use: empirical studies and quantitative models. *Ergonomics*, 43(7):931–951, July 2000.

[66] R. Parasuraman, T.B. Sheridan, and C.D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297, May 2000.

[67] J.C. Parrish, D.L. Akin, and G.G. Gefke. The ranger telerobotic shuttle experiment: Implications for operational eva/robotic cooperation. In *30th International Conference on Environmental Systems*, number 2000-01-2329. SAE International, 2000.

[68] K. Pilotte. Analysis of grasp requirements for telerobotic satellite servicing. Master's thesis, University of Maryland, 2004.

[69] P. Pina, ML Cummings, J.W. Crandall, and M.D. Penna. Identifying generalizable metric classes to evaluate human-robot teams. In *3rd Annual Conference on Human-Robot Interaction: Metrics for Human-Robot Interaction Workshop*, 2008.

[70] P. Pina, B. Donmez, and M.L. Cummings. Selecting metrics to evaluate human supervisory control applications. HAL Report HAL2008-04, Humans and Automation Laboratory, Massachusetts Institute of Technology, 2008.

[71] S.S. Ponda, H.L. Choi, and J.P. How. Predictive planning for heterogeneous human-robot teams. *AIAA Infotech@ Aerospace*, 2010.

[72] M. Prewett, R. Johnson, K. Saboe, L. Elliott, and M. Coovert. Managing workload in human-robot interaction: a review of empirical studies. *Computers in Human Behavior*, 26(5):840–856, September 2010.

[73] F. Rehnmark, R. Ambrose, and M. Goza. The challenges of extra-vehicular robotic locomotion aboard orbiting spacecraft. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2004.

[74] G. Rodriguez and C.R. Weisbin. A new method to evaluate human-robot system performance. *Autonomous Robots*, 14(2-3):165–178, 2003.

[75] F. Rohrmuller, O. Kourakos, M. Rambow, D. Brscic, D. Wollherr, S. Hirche, and M. Buss. Interconnected performance optimization in complex robotic systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4113–4118. IEEE, 2010.

[76] J.A. Saleh and F. Karray. Towards generalized performance metrics for human-robot interaction. In *Autonomous and Intelligent Systems, 2010 International Conference on*, pages 1–6. IEEE, 2010.

[77] H. Sato, H. Aguirre, and K. Tanaka. Controlling dominance area of solutions and its impact on the performance of moeas. In *Evolutionary Multi-Criterion Optimization*, pages 5–20. Springer, 2007.

[78] D.K. Saxena and K. Deb. Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3204–3211. IEEE, 2008.

[79] D.K. Saxena, T. Ray, K. Deb, and A. Tiwari. Constrained many-objective optimization: a way forward. In *Evolutionary Computation, 2009. IEEE Congress on*, pages 545–552. IEEE, 2009.

[80] J. Scholtz. Theory and evaluation of human robot interactions. In *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2002.

[81] J. Scholtz, B. Antonishek, and J.D. Young. Implementation of a situation awareness assessment tool for evaluation of human-robot interfaces. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4):450–459, July 2005.

[82] D. Schreckenghost, T. Fong, T. Milam, E. Pacis, and H. Utz. Real-time assessment of robot performance during remote exploration operations. In *Aerospace conference, 2009 IEEE*, pages 1–13. IEEE, 2009.

[83] D. Schreckenghost, T. Fong, H. Utz, and T. Milam. Measuring robot performance in real-time for nasa robotic reconnaissance operations. In *Proceedings of NIST PERMIS Workshop*, 2009.

[84] D. Schreckenghost, T. Milam, and T. Fong. Ai space odyssey: Measuring performance in real time during remote human-robot operations with adjustable autonomy. *IEEE Intelligent Systems*, 25(5):36–44, 2010.

[85] D. Schreckenghost, T. Milam, and T. Fong. Measuring performance in real time during remote human-robot operations with adjustable autonomy. *IEEE Intelligent Systems*, 25(5):36–45, 2010.

[86] J.A. Shah, J.H. Saleh, and J.A. Hoffman. Review and synthesis of considerations in architecting heterogeneous teams of humans and robots for optimal space exploration. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 37(5):779–, September 2007.

[87] J.A. Shah, J.H. Saleh, and J.A. Hoffman. Analytical basis for evaluating the effect of unplanned interventions on the effectiveness of a human-robot system. *Reliability Engineering and System Safety*, 93(8):1280–1286, 2008.

[88] S.M. Singer and D. Akin. *Role definition and task allocation for a cooperative EVA and robotic team.* Number 2009-01-2529. SAE International, 2009.

[89] S.M. Singer and D.L. Akin. Task scheduling for cooperative human/robotic space operations. In *International Conference on Environmental Systems*, number 2008-01-1985. SAE International, 2008.

[90] S.M. Singer and D.L. Akin. Analyzing the effect of robot task performance on scheduling a cooperative human and robotic team. *Acta Astronautica*, 66(1):102–116, 2010.

[91] S.M. Singer and D.L. Akin. A survey of quantitative team performance metrics for human-robot collaboration. In *International Conference on Environmental Systems*, number AIAA 2011-5248. AIAA, AIAA, 2011.

[92] B. Stanton, B. Antonishek, and J. Scholtz. Development of an evaluation method for acceptable usability. In *Proceedings of PERMIS*, 2006.

[93] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 33–40. Association for Computing Machinery, 2006.

[94] I. Tikach, A. Bechar, and Y. Edan. Switching between collaboration levels in a human-robot target recognition system. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):955–967, 2011.

[95] J.G. Trafton, N.L. Cassimastis, M.D. Bugajska, D.P. Brock, F.E. Mintz, and A. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(4):460–470, July 2005.

[96] E. Tunstel. Operational performance metrics for mars exploration rovers. *Journal of Field Robotics*, 24(8-9):651–670, 2007.

[97] A. van Wissen, Y. Gal, BA Kamphorst, and MV Dignum. Humanagent teamwork in dynamic environments. *Computers in Human Behavior*, 28(1):23 – 33, 2012.

[98] H. Wang, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara. How search and its subtasks scale in n robots. In *Proceedings of the 4th ACM/IEEE International conference on human robot interaction*, 2009.

[99] J. Wang and M. Lewis. Assessing cooperation in human control of heterogeneous robots. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, March 2008.

[100] D.J. Weigel, R.S. McDaniel, and J.V. Thornton. Eva checklist: Sts-103 flight supplement. Technical Report JSC-48024-103, NASA Johnson Space Center, 1999.

[101] Y.G. Woldesenbet, G.G. Yen, and B.G. Tessema. Constraint handling in multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 13(3):514–525, June 2009.

[102] R.E. Yagoda and M.D. Coovert. How to work and play with robots: an approach to modeling human-robot interaction. *Computers in Human Behavior*, 28(1):60–68, 2012.

[103] H.A. Yanco and J.L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, 2002.

[104] H.A. Yanco, J.L. Drury, and J. Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction*, 19(1):117–149, 2009.

[105] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), 1999.

[106] E. Zitzler. Hypervolume metric calculation, 2001.

[107] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Parallel Problem Solving from Nature - PPSN-V*, pages 292–301, 1998.

[108] E. Zitzler and L. Thiele. multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[109] X. Zou, Y. Chen, M. Liu, and L. Kang. A new evolutionary algorithm for solving many-objective optimization problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(5):1402–1412, 2008.