9-2015

Real-Time Supervision for Human Robot Teams in Complex Task Domains

Arif Tuna Ozgelen
Graduate Center, City University of New York

How does access to this work benefit you? Let us know!
Follow this and additional works at: https://academicworks.cuny.edu/gc_etds

Recommended Citation

This Dissertation is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2015
This manuscript has been read and accepted by the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Professor Elizabeth Sklar

Date
Chair of Examining Committee

Professor Robert Haralick

Date
Executive Officer

Professor Simon D. Parsons
Professor Jizhong Xiao
Professor Michael Lewis
Supervisory Committee
Abstract

Real-Time Supervision for Human-Robot Teams in Complex Task Domains

by

Arif Tuna Özteken

Adviser: Professor Elizabeth Sklar

Ongoing research on multi-robot teams is focused on methods and systems to be utilized in dynamic and dangerous environments such as search and rescue missions, often with a human operator in the loop to supervise the system and make critical decisions. To increase the size of the team controlled by an operator, and to reduce the operator’s mental workload, the robots will have to be more autonomous and reliable so that tasks can be issued at a higher level. Typical in these domains, such high-level tasks are often composed of smaller tasks with dependencies and constraints. Assigning suitable robot platforms to execute these tasks is a combinatorial optimization problem. Operations Research and AI techniques can handle large numbers of robot allocations in real time, however most of these algorithms are opaque to humans; they provide no explanation or insight about how the solution is produced. Recent studies suggest that interaction between the human operator and robot team requires human-centric approaches for collaborative planning and task allocation, since black-box solutions are often too complex to examine under stressful conditions and are often discarded by experts.

The main contribution of this thesis is a methodology to help operators make decisions about complex task allocation in real time for high stress missions. First a novel, human-centric graphical model, Task Assignment Graph (TAG), is described to analyze and predict the complexity of task assignment and scheduling problem instances, taking into account the spatial distribution of resources and tasks. Then, the TAG model is extended for dynamic environments to the Mission-level Assignment Problem (MAP) model. Two user studies were conducted, first in static and then in dynamic environments, in order to identify and empirically verify the key factors, derived from the graphical model, which affect the decision making of human supervisors during task assignment for a team of robots. In these user studies, participants used software tools developed for this work.
One of these software tools allows for two different levels of autonomy for the interaction scheme: manual control and collaborative control, with an option to invoke an automated assignment tool. Findings relating to the impact of decision support functionality on the mental workload and the performance of the supervisor are presented. Finally, steering of the common algorithms utilized by decision support tools, using the strategies employed by user study participants, related to the TAG and MAP model parameters, are discussed.
Acknowledgments

First of all I’d like to thank my advisers, Prof. Elizabeth Sklar, who is a mentor to me and kept me on the path to completion no matter how hard I tried to diverge and Prof. Simon Parsons, whose invaluable insights and comments shaped the direction of my research. Their continuous support and guidance helped me tremendously. I’d also like to thank my committee members Prof. Jizhong Xiao and Prof. Michael Lewis for their valuable input. Among many faculty members in The Graduate Center, I’d like to specially thank Prof. Susan Epstein, whom I worked with on HRTeam project, for her support and all that she have taught me.

During my studies, I’ve met and worked with many people whom I had the pleasure of calling my friends as well as my colleagues. I’d like to thank all members of the Agents research group, especially to Sumon Azhar and Artyom Diky whom I worked, traveled and shared many good memories during this chapter of my life. Also I’d like to thank Michael Kaisers, for always keeping me in check and for many intellectually stimulating conversations.

In some of my experiments I used HRTeam framework, which is a result of a collaborative effort. I’d like to thank all the members of HRTeam and Metrobotics projects for their valuable work. Among many people who participated in these projects, I’d like to specially thank Eric Schneider, Michael Constantino and Ofear Balas who went out of their way to help me in many occasions and to Farah Abbasi, who designed the initial prototype of the user interface.

Biggest challenges that I faced during my pursuit of a doctoral degree was motivational. The moral support that all my extended family have given, gave me strength to continue during tough times. I’d like to specially thank my cousin, Arda Nural, for constantly pushing me to finish my dissertation (“Ne zaman bitiyo olm?”). Finally, without the encouragement and unconditional support of my mother Ufuk Ö zgelen and my wife Burçin Ö zgelen, this work would not have been
possible. My gratitude for them is beyond words. Dear Burçin, you have been with me and beside me at every step. Please know that this work is a result of your moral and emotional support. I hope this work to be an example to our son Kağan Bora and a motivation to push himself beyond his limits. Dear Mother, you have given me everything to help me to get to this point and to be who I am today. This dissertation is dedicated to you.
## Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and Scope</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Research Questions and Statement</td>
<td>8</td>
</tr>
<tr>
<td>1.3 Approach</td>
<td>9</td>
</tr>
<tr>
<td>1.4 Contributions</td>
<td>10</td>
</tr>
<tr>
<td>1.5 Thesis Outline</td>
<td>11</td>
</tr>
<tr>
<td>2 Background</td>
<td>13</td>
</tr>
<tr>
<td>2.1 Human-Robot Interaction</td>
<td>13</td>
</tr>
<tr>
<td>2.1.1 Evaluation of HRI systems</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Multi-Robot Task Allocation (MRTA)</td>
<td>24</td>
</tr>
<tr>
<td>2.2.1 Market-based Approaches</td>
<td>24</td>
</tr>
<tr>
<td>2.2.2 Constraint Programming (CP) Techniques</td>
<td>27</td>
</tr>
<tr>
<td>2.2.3 Algorithm Steerability</td>
<td>32</td>
</tr>
<tr>
<td>2.3 Related Work</td>
<td>35</td>
</tr>
<tr>
<td>2.4 Thesis Scope</td>
<td>39</td>
</tr>
<tr>
<td>3 MRTA Complexity Analysis in Static Environments</td>
<td>43</td>
</tr>
</tbody>
</table>
### 3.1 Overview of Task Taxonomies ........................................... 44
### 3.2 Task structure in Multi-Robot Systems ............................... 45
### 3.3 Task Assignment Graph (TAG) .......................................... 48
### 3.4 Task Complexity Assessment Tool (TCAT) Test bed .................. 51
### 3.5 TCAT User Study ................................................................ 54
  - 3.5.1 Experimental Setup ..................................................... 55
  - 3.5.2 Procedure ................................................................. 56
  - 3.5.3 Metrics and Variables .................................................. 58
  - 3.5.4 Results and Discussion ............................................... 61
  - 3.5.5 Post-experiment Interviews .......................................... 67
### 3.6 Summary ............................................................................. 69

### 4 MRTA Complexity Analysis in Real-Time and Dynamic Environments 71

#### 4.1 Extending TAG for Real-Time and Dynamic Environments .............. 73
  - 4.1.1 Metrics ................................................................. 76
  - 4.1.2 MAP adaptation into TASC experiment ........................... 79

#### 4.2 Task Assignment Supervisory Control (TASC) Interface and Testbed 80
  - 4.2.1 Manual Allocation Mode .......................................... 81
  - 4.2.2 Automated Allocation Mode ....................................... 82
  - 4.2.3 Collaborative Allocation Mode .................................... 84

#### 4.3 HRTeam Framework ........................................................ 85

#### 4.4 TASC User Study ............................................................. 87
  - 4.4.1 Experimental Setup ..................................................... 89
  - 4.4.2 Procedure ................................................................. 91
  - 4.4.3 Metrics and Variables .................................................. 92
  - 4.4.4 Results and Discussion ............................................... 93

#### 4.5 Summary ............................................................................. 102

### 5 Mission-level Analysis and Integration of MAP model into MRTA methods 104

#### 5.1 Mission-level analysis of TASC experiment ............................... 105
  - 5.1.1 Analysis of Mission-level Objective Metrics ....................... 107
5.1.2 Analysis of Mission-level Subjective Metrics ........................................ 110
5.1.3 Analysis of Interaction and Plan Execution Data ................................. 112
5.1.4 Analysis of Collaborative Control Functionality .................................. 114
5.1.5 Summary ......................................................................................... 116
5.2 Post-experiment interviews ................................................................. 118
5.2.1 Usability and Task Complexity ......................................................... 119
5.2.2 Strategies employed in TCAT and TASC for task assignment ............ 122
5.2.3 Collaborative Allocation Functionality .............................................. 125
5.3 Integration of TASC heuristics into Autonomous Planners ..................... 126
5.4 Summary ......................................................................................... 132

6 Conclusion ......................................................................................... 134
6.1 Revisiting Research Statement and Questions ..................................... 134
6.2 Contributions ................................................................................... 137
6.3 Limitations of the Methodology and the Models ................................. 140
6.4 Future work ..................................................................................... 142
6.4.1 The Models .................................................................................. 142
6.4.2 Decision Support ......................................................................... 143
6.4.3 Problem Domain ......................................................................... 144
6.5 Summary ......................................................................................... 145

Appendices .......................................................................................... 146
A Task Analysis .................................................................................... 147
B TCAT User Study Procedure ............................................................. 152
B.1 Instructions ..................................................................................... 152
B.2 Post-Experiment Interview Questions ............................................. 154
C TCAT Scenarios ................................................................................ 155
C.1 Category Q1 - CTR low, APR low ...................................................... 155
C.2 Category Q2 - CTR high, APR low ................................................... 158
# List of Figures

2.1 *Independent System Effectiveness* .................................................. 22  
2.2 *Dependent System Effectiveness* ..................................................... 22  
2.3 *Example: sub-optimal single-item auction allocation* .......................... 26  

3.1 *TCAT scenario example 1* .............................................................. 51  
3.2 *TCAT scenario example 2* .............................................................. 52  
3.3 *TCAT Interface* .............................................................................. 53  
3.4 *TCAT Map Topologies* ..................................................................... 56  
3.5 *Mental demand* .............................................................................. 63  
3.6 *Perceived performance* .................................................................... 63  
3.7 *Task assignment completion times* .................................................... 64  
3.8 *Objective performance score* ........................................................... 65  

4.1 *A TAG, where robots* \( r_1 \) and \( r_2 \) are working on task* \( t_4 \) .............................................. 76  
4.2 *A consecutive TAG, after robots* \( r_1 \) and \( r_2 \) have completed task* \( t_4 \) *in Figure 4.1* .......................... 77  
4.3 *A TAG after the situation in Figure 4.2, where a new task* \( t_7 \) *appears.* ........................................ 78  
4.4 *TASC Interface* .............................................................................. 81  
4.5 *Auto assign algorithm* .................................................................... 84  
4.6 *HRTeam Architecture. Reprinted from [89]* ....................................... 86  
4.7 *Objective Mental Demand Analysis* ................................................... 95  
4.8 *Objective Performance Score Analysis* .............................................. 95  
4.9 *Subjective Workload Analysis* .......................................................... 96  
4.10 *Subjective Performance Analysis* ..................................................... 97
List of Tables

2.1 Sheridan's LOA ....................................................... 14
2.2 Hearst Interaction Scale ........................................... 14
3.1 TCAT experiment scenario categories ......................... 57
3.2 TCAT Dependent Variable Statistics ......................... 60
3.3 TCAT Participant Demographics ............................... 61
3.4 TCAT Summary Statistics ........................................ 61
3.5 Pairwise T-tests .................................................... 67
4.1 TASC Participant Demographics ............................... 89
4.2 TASC Scenario Properties ......................................... 91
4.3 Incomplete Plans distributed by TAG ID ..................... 94
4.4 TASC Summary Statistics ........................................ 97
4.5 TASC Summary Statistics grouped by TAG ID ................ 101
5.1 Mission-level Time metrics ..................................... 109
5.2 Mann-Whitney-U test results for objective metrics ........ 109
5.3 NASA-TLX metrics summary .................................. 110
5.4 Mann-Whitney-U test results for subjective metrics (NASA-TLX) 112
5.5 Mission-level Interaction metrics: Button-clicks and Keystrokes 112
5.6 Mission-level Plan Execution Metrics ....................... 114
A.1 Task classification analysis .................................... 147
Acronyms

ADD  Average Domain Density. 78, 79, 89, 94–97, 102, 136, 143

APR  Average Platform Requirement. 50, 54, 56, 57, 59, 61–65, 69, 72, 76, 102, 136, 138

COP  Constraint Optimization Problem. 27, 33, 34, 127

CP   Constraint Programming. 105, 126–130, 133

CSP  Constraint Satisfaction Problem. 27–31, 33, 34, 127, 128


DCOP Distributed Constraint Optimization Problem. 30, 31, 127

DSCP Distributed Constraint Satisfaction Problem. 29–31, 127

E-GAP Extended Generalized Assignment Problem. 40, 135

FO   Fan Out. 3

GAP  Generalized Assignment Problem. 40, 88, 135

HMRI Human-Multi-Robot Interaction. 10, 32, 41

HRI  Human-Robot Interaction. 1, 2, 8–11, 13–15, 18, 19, 21, 23, 33, 47, 85, 127, 135, 137, 138

LOA  Level of autonomy. 13, 14, 41, 82, 85
MAP Mission-level Assignment Problem. iii, iv, 74, 75, 77, 79, 87, 88, 102, 105, 126, 133, 136, 138, 140, 142, 143, 145, 146

MAS Multiagent Systems. 1, 2, 13, 24, 25, 27–31, 43, 46, 69, 74, 113, 144

MRS Multi-Robot Systems. 2, 24, 25, 31, 40, 43, 48, 69, 74, 85, 113, 137–139, 142, 144, 146

MRTA Multi-Robot Task Allocation. 13, 27, 32, 69, 82, 83, 105, 119, 127, 133, 137, 139, 144, 146

NASA-TLX NASA Task Load Index. 18, 58, 85, 92, 93, 98, 100, 103, 105, 107, 110, 112, 116, 120, 132

PSI Parallel Single-Item Auctions. 27

SOMR Single-Operator Multi-Robot. 2, 3, 24, 32, 126, 138, 146

SSI Sequential Single-Item Auctions. 27, 34, 82–84, 130


TCAT Task Complexity Assessment Tool. 10, 43, 52, 66, 69, 73, 76, 80, 92, 93, 99, 100, 102, 104, 119, 120, 122–124, 132, 133, 136, 138, 141, 145, 155

TDR TAG Disruption Ratio. 78, 79, 89, 94–98, 102, 136

UAV Unmanned Aerial Vehicle. 1, 5, 6, 41, 139

UGV Unmanned Ground Vehicle. 11, 139

USAR Urban Search and Rescue. 69, 141, 142, 144

USV Unmanned Surface Vehicle. 5

UV Unmanned Vehicle. 1, 3, 7, 39
Chapter 1

Introduction

Robotics has been a highly active area of research in recent decades. Along with the successful utilization of robots in the manufacturing industry, mobile robotics became the next frontier and have received significant attention from the research community. Application areas of mobile robots range from everyday tasks in common environments such as home cleaning (e.g., iRobot roomba) and entertainment (e.g., Sony AIBO) to critical missions in extreme environments such as space exploration (e.g., NASA Mars rovers) and military applications (e.g., Unmanned Aerial Vehicle (UAV)). Considerable research effort was dedicated to utilization of robot teams, where overall performance and effectiveness of the task may potentially benefit from parallel, coordinated execution (e.g., urban and wilderness search and rescue). Such extreme and critical situations often require accountability and human expertise. The problem investigated in this dissertation focuses on human-in-the-loop systems for controlling multirobot or Unmanned Vehicle (UV) teams in these extreme environments and critical mission domains.

The theoretical foundations for the work presented in this document are derived mainly from two fields: Multiagent Systems (MAS) and Human-Robot Interaction (HRI). The MAS field is primarily concerned with building systems containing multiple intelligent agents that are capable of autonomous action in order to achieve their design objectives and capable of interacting with other
agents [99, 87]. MAS applications are potentially suitable for complex real-life problems, which are
distributed in nature. As opposed to centralized approaches, which often suffer from computational
complexity and limited scalability, the MAS field is concerned with designing societies of self-
interested agents, each concerned with a part of the problem and coordinating or cooperating with
other agents to improve the global solution whenever it suits their own individual interest. As Lesser
states [56], the key to a successful MAS application lies in the efficient cooperation and coordination
among its individual agents. Although this approach doesn’t guarantee globally optimal solutions,
it provides scalable and robust solutions for a set of problems. MAS has been successfully applied
to multirobot systems, which is the scope of this dissertation.

Multi-Robot Systems (MRS) have been investigated under MAS as a sub-field, where agents
are robots and they share a common goal, therefore assumed to be benevolent. Although this
assumption is reasonable, due to possible disparity among robots’ perceptions of the environment
and also the uncertainty in the physical world in general, team coordination still poses a significant
challenge. Therefore solutions sought by the MAS community to problems arising from team
coordination, such as distributed reasoning, planning and task/resource allocation, are also relevant
to MRS applications. Research on these problems, proposed solutions and approaches have inspired
and formed the basis of this dissertation. Application domains frequently studied in MRS, such
as multi-robot exploration, generally involve simple, independent tasks of directing each robot to
a desired location. More interesting and realistic cases in MRS, as considered in this work, lie
in the realm of applications where tasks have time constraints and/or dependencies among them
[15, 23, 25, 24, 54, 9].

The other field of research, to which thesis is primarily contributing, is Human-Robot
Interaction. HRI is a multidisciplinary field focusing on all forms of interaction between humans
and robots. Enabling humans to control multiple robots is one of the goals of HRI and is referred
to as Single-Operator Multi-Robot (SOMR) control [36, 96, 88]. SOMR control methods have been
investigated in environments where robots are expected to coordinate through division of labor on a set of independent tasks, such as urban and wilderness search and rescue [97, 96, 39], as well as in environments where tasks require tight coordination, such as robotic assembly [42, 43]. Regardless of the domain, SOMR control methods have been studied mostly in environments where increasing the numbers of robots could potentially improve the performance of the team. The main motivation behind SOMR research stems from the necessity of involving human experts or operators in critical missions. Due to sensory limitations and the autonomous capabilities of robots, expecting robots to operate robustly in these domains is not realistic at the moment or in the near future [67]. Also, human-in-the-loop team structures are potentially very useful in situations where a fast decision based on an operator's experience can save a lot of computation time. As the autonomous capabilities of UVs increase, it is expected that the ratio of the number of operators to UVs will decrease, allowing operators to command a team of such platforms [45]. As a result, interactions between the operator and the robot team will place more emphasis on supervisory and higher-level strategic commands rather than low-level control.

Earlier studies in SOMR research focused on playbook-style management, sequential control of each robot via teleoperation or simple navigational commands [26, 97, 96]. Research areas in SOMR include finding a suitable level of autonomy for the team that would improve performance [35, 39, 42, 43, 97] and defining metrics to measure this performance in order to compare different interaction mechanisms [67, 68, 81]. In domains where operators are able to interact at a higher level, the number of UVs that an operator needs to control simultaneously, referred to as Fan Out (FO) [68], is limited. To reduce the cognitive workload and prevent loss of situational awareness [27], integration of automated planners in control systems to collaborate with or support human operators, is also investigated.

In practice, one of the limitations of utilizing automated task assignment and planning systems for decision support or collaborative control is cognitive on the part of the human operator.
Plans devised by state-of-the-art systems are often discarded by human experts if they are not fully understood or conflict with users' expectations [49]. Also, such systems often work using specifications defined during their design [10], resulting in limited and restricted interaction with operators who did not participate in the design phase.

So far these systems have not been successful in maintaining situational awareness of the operators. Reasons for the failure is attributed to lack of plausible models that capture the operator’s perspective of the mission and the design of automated task planners which often provide black-box solutions and are unresponsive to operator expectations, mental workload and strategies. In addition, supervisory control has mostly been studied for cases with simple and independent tasks, which is not a realistic assumption for potential applications of these systems.

This thesis introduces a human-centric model that captures task assignment complexity as perceived by humans. The assignment complexity is represented as a combination of resource availability and distribution in the spatial sense, as well as the characteristics and requirements of the tasks. The intent is to be able to detect operator mental workload and situational awareness changes and provide information for steering algorithms commonly used by automated planning systems (e.g., auction mechanisms, constraint solvers, etc.) towards operator expectations, thus mitigating the loss of operator situational awareness. As opposed to existing work, the model presented in this thesis considers complex tasks that may require multiple robots and may depend on other tasks due to spatial restrictions imposed by the environment.

My models are evaluated by conducting two user studies. The first study considers only static environments where the users are not under time pressure, in order to verify the key factors of the model. The second study considers dynamic environments where the subjects are under time pressure, in order to study the mental workload and actions of human users in situations similar to real-life scenarios. The latter study also investigates the affect of utilizing a decision support system on the perceived performance and the mental workload of the users. Lastly, common strategies
employed by the subjects in these user studies are investigated, followed by a discussion about how these strategies can be used to guide autonomous decision support systems, with hopes to benefit designers of similar systems in the future.

1.1 Motivation and Scope

Dull, dangerous and dirty environments are frequently suitable candidates for robotic applications. Currently, search and rescue domains and military operations are the focus of research that parallels the work presented in this thesis. Utilization of robots in these environments, although in a limited fashion, is already a reality.

One of the first known uses of robots in search and rescue operations was at the World Trade Center (WTC) after the 9/11 attacks, discussed in detail by Murphy [63, 64]. The robots were used mainly for searching for victims, searching for paths that would be quicker to excavate debris, inspecting structural damage and detecting hazardous materials. Similarly, during the recovery efforts after the Japan earthquake (2011), robot teams were sent to search for survivors in the rubble [37]. Robots were also used in the aftermath of the 2010 Gulf of Mexico oil spill, to activate a device that is designed to shut off a well in the event of a sudden pressure release [74]. Ongoing work by Scerri et al. [77] aims to utilize a team of Unmanned Surface Vehicle (USV) for flood disaster mitigation.

Human-robot teamwork is also studied in the context of military applications. Carlin et al. [9] describe an initial study for supervising robots to help soldiers for room-clearing missions. Jacobs et al. [45] describe the current state of the Unmanned Air Vehicles (UAVs) program in the United States Air Force (USAF) and state that one of the main problems is that these platforms are manned by pilots whose training is costly, intensive and time consuming. General Phillip M. Breedlove states, “The No. 1 manning problem in our Air Force is manning our unmanned platforms” [78]. The proposed approach by the community to overcome this problem is the introduction of intelligent
aiding tools for a new type of operator, who is not a pilot and is expected to interact with the UAVs at a supervisory level.

Robots are being used in industrial applications as well. Breitenmoser et al. [8] describe their deployment of climbing robots to inspect power plants for aging, corrosion and mechanical stresses that could cause structural damage. Due to lack of trained personnel to carry out inspections, the authors suggest the system should have high levels of autonomy.

In all of the above scenarios, current systems require extensive training of human operators, which is costly and time consuming. Therefore, expecting to be able to man every single platform for large-scale deployments is unrealistic. The solution lies in improving levels of autonomy of the robots and empowering operators with the option to provide high-level supervisory input, as opposed to micro-management. In the current state of technology, robots are capable of handling independent tasks autonomously (such as navigation), but only under certain circumstances. When the environment is as noisy and cluttered as the WTC site [63, 64], even teleoperation is very hard for human operators, let alone supervising a team of robots. However, if the environment is suitable for a higher level of autonomy for navigation and other tasks, then it is reasonable to deploy a larger number of robots under the control of a single operator, that is, if the overall mission can benefit from more robots.

The assumption made in this thesis is that the autonomous capabilities of robots will improve further in the coming years, judging by the progress of research and the level of support from governments and various industries for the field of robotics. In parallel to this progress, the expectation is that robots will evolve towards handling tasks that are interdependent and time constrained. Therefore, models of interaction with humans will become more important for supervisory control. There is already some work in this context that uses UAV platforms [23, 25, 24, 54].

Regardless of the platform, so far, the solutions for human-robot interaction are domain
specific, informal and mostly approached at the design level. Formal models and frameworks for teamwork are developed mostly for autonomous operations, with algorithmic efficiency and scalability as the main goals, while human factors are proving to be the major obstacle as task execution becomes more complex. To overcome the problems posed above, formal methods of interaction and evaluation have to be addressed.

My work studies factors affecting the decision-making process of a human operator when supervising multiple UVs in a complex task environment. The environments I investigate are considered complex largely because multiple tasks must be accomplished, by one or more robots, and the tasks have different types of dependencies on one another.

The motivation to develop a human-centric model of such environments is based on several reasons. First, it is imperative to understand the human factors that contribute to decision making in order for human operators, working under stressful conditions, to be able to interact effectively with automated task assignment and planning systems. My theory is that utilization of a more accessible model may improve the design of such automated systems particularly in areas such as: deciding when to take the initiative; deciding when to interrupt the operator; predicting degradation in the operator’s situational awareness, before it impacts mission performance; and algorithm steering [3] for planning based on operator expectations. Also, a human-centric model may provide context for explaining plans to human operators, for example using natural language or argumentation-based dialogue. Lastly, studying human behaviors that arise during task assignment problems could highlight trends (such as solving certain types of sub-problems first) that could be used for improving the performance of agents that assist humans in addressing such problems, as well as improving training experiences for operators.
1.2 Research Questions and Statement

The thesis of the work presented in this dissertation is that human-centric approaches for assisting the operator in human multi-robot teams for task and resource allocation can mitigate some circumstances in which performance would otherwise degrade due to declining situational awareness.

In order to verify this statement, first a model of factors that influence the human decision-making process for task and resource allocation is built and verified. Then, based on this model, steering or guiding commonly used algorithms and techniques is investigated, in order to improve the interaction between these autonomous components of the system and the humans. My hypothesis is that the steering of the automated components can best be achieved by developing these components such that they encompass strategies commonly used by human users.

As stated previously, the domain under investigation contains tasks with inter-dependencies and may require a coalition of robots for execution. Furthermore, I envision a dynamic environment where new tasks will be introduced over time, hence may require replanning during execution. Under the assumption of these domain characteristics, the proposed work addresses a number of research questions.

At the highest level, the research questions under investigation are mostly related to the effectiveness and the nature of human-agent collaboration or decision support:

**RQ1** Can an intelligent HRI system be of measurable help to a human operator, when making decisions regarding allocation of tasks with constraints and dependencies?

**RQ2** How can an intelligent HRI system be guided according to operator expectations when reasoning about task/resource allocations in real-time, during high-stress missions?

To address both RQ1 and RQ2, first, a model is required that represents the general characteristics of human cognitive processes involving decision making during supervision of a robot team. Questions related to this model that are addressed in this work are as follows:
RQ3  What are the factors and to what degree do they affect a human supervisor’s decision making regarding task assignment under stress-free and favorable circumstances?

RQ4  What are the factors and to what degree do they affect a human supervisor’s decision making regarding task assignment during real-time high-stress missions?

Once the model is verified and the key factors are determined, the question of how to integrate this knowledge into the existing autonomous task allocation algorithms is addressed, in order to design better interaction mechanisms, which leads to the following research questions:

RQ5  What are the common strategies adopted by the supervisors?

RQ6  How can useful human strategies be integrated with common MRTA techniques?

1.3  Approach

The general approach taken in this thesis involves:

1. an in-depth study of the expected task properties in future applications of human multi-robot teams,

2. identification of the key factors affecting operators’ decision making,

3. building a model for analyzing the situation from the operator’s perspective,

4. verification of the model, and

5. providing a guideline for building an autonomous component to a decision support system that replicates commonly employed strategies by humans, in order for human operators to steer the solution in a desired direction.

Verification of the model is done empirically, which is a common approach in the HRI field, because cognitive processes involved in decision making under the domain studied are not fully understood.
The verification and identification of the key components of the model are done in iterative steps. The first step is a user study where participants were presented scenarios and expected to assign robots to tasks and rate the complexity of the scenario and their performance, subjectively. For this user study, participants used a software tool, the Task Complexity Assessment Tool (TCAT), developed for this user study. By design, the experiment focused on the task allocation phase, presenting a large number of short scenarios and the robots were not expected to follow through with executing the allocated tasks.

The second step extends and uses the model during real-time missions where the tasks are executed by simulated robots and new tasks are generated dynamically. To this end, a multirobot test environment, the HRTeam framework, and a supervisory control interface, the Task Assignment Supervision and Control (TASC) tool, are used. Although the HRTeam framework allows running experiments both in simulation and with physical robots, in this thesis all experiments were conducted in simulation. The TASC tool was developed specifically for this thesis, and allows supervisors to perform manual task assignments, use decision support or work in a fully autonomous mode, which is used for assessing operator performance.

1.4 Contributions

The main contributions of this work are primarily to the field of HRI. The experiments and the methodologies explained in the remainder of this thesis are mostly within a sub-domain of this field, Human-Multi-Robot Interaction (HMRI).

The fundamental research goal of this work is to investigate approaches that incorporate multiagent task/resource allocation methods together with human-centric methods to improve interaction and assistance to a human operator with task assignment, and therefore reduce mental workload.

To my knowledge, the proposed work is a novel human-centric approach for allocation of
complex tasks for Unmanned Ground Vehicle (UGV)s and a step towards generating a method of interaction for supervisory control of multiple ground robots in such domains. The contributions are as follows:

1. Investigation of domain characteristics in a principled manner is required for acquiring generalizable results that are derived from any HRI study. To this end, a task taxonomy is devised from the human’s perspective, as opposed to existing taxonomies which are mostly from the computational perspective.

2. A detailed study of factors affecting the complexity of the task allocation, such as the class of problem based on our taxonomy and the state of the environment. This will provide us with a method to analyze and compare the quality of the allocation of particular instances with others.

3. An in-depth analysis of human behavior and strategies employed during multi-robot team supervision, and their integration into common autonomous task allocation algorithms, for improved decision support functionality.

4. Finally, production of an empirical data set of a multi-robot system, describing human operators’ real-time reactions when dealing with complex tasks. This data is valuable since user studies in multi-robot teamwork are hard to acquire, and the results will be generalizable due to the detailed specification of the environmental and task characteristics based on the aforementioned task taxonomy and complexity studies.

1.5 Thesis Outline

The remainder of this dissertation is organized as follows:

- In Chapter 2, some of the related work on HRI and multi-robot task allocation are discussed in detail and the differences between the related work and my approach are highlighted.
• Chapter 3 covers the structure of tasks commonly studied in the literature and factors affecting the complexity of a task assignment problem. A graph-based model of spatial distribution of tasks and resources is presented (TAG), followed by the results of a user study that verifies the model parameters in static environments without any induced time pressure on the human subjects.

• Chapter 4 describes the expanded version of the graph-based model introduced in Chapter 3, and extends the list of factors for real-time and dynamic scenarios. Results of a user study, conducted with real-time assignment missions, is presented. The study involves users interacting with the system in two modes of operation: manual and collaborative. Results include reports on factor verification, objective and subjective performance.

• Chapter 5 presents observations made on common strategies employed by the users in the study described in Chapter 4 and discusses how they can be incorporated into an autonomous system. Finally, along with other well-known heuristics used in MRTA, these strategies are compared in terms of performance and computational complexity.

• The final chapter recapitulates the thesis statement and discusses how the findings from the experiments support my claims, states limitations of the model and important open questions and finally point out the future directions of this line of research.
Chapter 2

Background

This chapter provides some background information mainly in the two areas that this dissertation derives from: Human-Robot Interaction (HRI) (Chapter 2.1) and Multi-robot Task Allocation (Chapter 2.2). While work in HRI guides this research in terms of application domains, common metrics and evaluation techniques used in the field of Multi-Robot Task Allocation (MRTA), extending from the MAS field, provide formal computational approaches, methods and algorithms to address similar problems. An overview of selected and recent research that is found to be related to this work is presented in Chapter 2.3 followed by the thesis scope in Chapter 2.4, where the differences between this work and these selected works will be highlighted.

2.1 Human-Robot Interaction

The main concern of HRI is to understand and shape the interactions between one or more humans and one or more robots [36]. From the perspective of this work and the problem requirements examined in this dissertation, the components of HRI that affect the interaction are the autonomous capabilities of the robots, nature of information exchange and the team structure.

Level of autonomy (LOA) is a term used to reflect the autonomous capabilities of the mobile robots and indirectly refers to the amount of time that a robot can be neglected. In HRI, autonomy
is not the end goal and it is useful as long as it supports interaction. The notion of LOA is the
most strong human-centric application of the autonomy concept. An earlier and widely adopted
LOA description is Sheridan’s [86] (see Table 2.1) in the context of Human-Computer Interaction.

| not autonomous | 1. Computer offers no assistance; human does it all. |
|                | 2. Computer offers a complete set of alternatives. |
|                | 3. Computer narrows selection down to a few choices. |
|                | 5. Computer executes an action if human approves. |
|                | 6. Computer allows the human limited time to veto before automatic execution. |
|                | 7. Computer executes automatically and then necessarily informs the human. |
|                | 8. Computer informs human after automatic execution only if human asks. |
|                | 9. Computer informs human after automatic execution only if it decides too. |

Table 2.1: Sheridan’s Level of Autonomy (LOA) from [86]

Hearst [41] defined a scale for LOA from the HRI point of view (see Table 2.2). Hearst’s
LOA focuses more on the interaction between the agents and is a flexible interaction strategy based
on the context.

To make the interaction beneficial, the manner by which the information is exchanged
is vital. Measures of efficiency of interaction include interaction time required for intent and
instructions, cognitive and mental workload of an interaction, the amount of situation awareness
produced by the interaction, and the amount of shared understanding or common ground between
humans and robots.

In HRI problems where more than one human and/or robot exists, the structure of the team
becomes an important factor in interaction. A question that has received considerable attention
is how many robots a single human can manage. The answer depends on the LOA, the task and
the available modes of communication. Mitchell and Cummins [15] assert that given sophisticated

| direct control | Teleoperation |
|               | Mediated teleoperation |
|               | Supervisory control |
|               | Collaborative control |
| dynamic autonomy | Peer-to-peer collaboration |

Table 2.2: Interaction scale by Hearst from [41]
autonomy, it is possible for a human to control multiple robots in certain domains. Team organization and structure also have key effects on interaction. Questions such as who has the authority to make certain decisions, how the conflicts are resolved, and how roles are defined, and whether roles change during the course of a mission have important effects on the interaction. In many existing and envisioned problems, HRI will include not only humans and robots, but also software agents, for example, in the form of an intelligent interface, or agents acting as mission coordinators [7].

Following this guideline, in this dissertation, the HRI systems considered are expected to interact with the human operator in 4 different levels (2-5) according to Sheridan’s scale (see Table 2.1) and specifically focus on supervisory control in Hearst’s (see Table 2.2). The information exchange medium is generally a graphical user interface (GUI) in such systems and the context of communication is task assignment, which is one of the key foci of the presented work in this thesis and will be discussed at length in Chapters 3 and 4.

2.1.1 Evaluation of HRI systems

The human cognitive process is not fully understood and due to the infancy of human multi-robot applications, evaluation of HRI systems, especially in this domain, is hard and requires careful consideration. There are no standard measures, but there are sound guidelines which researchers in the field try to follow.

For accurate evaluation, two stages of interaction have to be distinguished. These two stages, as stated by Scholtz [81], are: the *perceptual* stage, where an operator tries to grasp the situation, and the *intervention* stage, where the operator gives commands to the system. The latter stage is related to the functionality and usability of the interaction medium and in most cases, can be measured more easily than the perceptual stage, which occurs mostly in an operator’s mind.

A commonly used measure, *situational awareness*, becomes critical during the evaluation of the perceptual stage since it captures not only basic information such as positions of robots, but
also a representation of the knowledge of the user about what is going on around him. Adams [2] states that the most commonly accepted definition of situational awareness is provided by Endsley [27]:

_The perception of elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future._

This definition incorporates three levels of situational awareness:

- **(Level 1) Perception:** Represents perception of the environment relative to the assigned mission, including the environmental status, attributes, and dynamics. Humans rely on visual, auditory etc. senses for environmental perception.

- **(Level 2) Comprehension:** Humans achieve level 2 situational awareness by integrating their environmental perceptions (level 1 situational awareness) with their goals and associated information from memory. It has been shown that this level of situational awareness can be improved by additional training and experience resulting in a mental model of task, thus improving the humans’ ability to understand the task.

- **(Level 3) Projection:** Humans are able to predict what will occur in the near future based upon their perception and comprehension of the situation, thus level 3 situational awareness is directly dependent upon attaining good level 1 & 2 situational awareness. Projection requires an excellent understanding of the mission domain and is frequently a highly demanding cognitive activity. Various aspects such as cognitive workload, mental capacity, and environmental stressors can limit humans’ level 3 situational awareness.

Human situational awareness is affected by stress, training, capabilities, human error, communication errors and task complexity. Due to limited perception and memory, humans often rely on timing, expectation and mental models. Timing and expectations are critical for guiding perceptual attention, comprehension and projection. While inaccurate expectations lead to
misunderstandings and inaccurate projections, timing of events affects the temporal component of situational awareness. To make up for the limitations associated with memory, mental models, scripts and heuristics are employed. Mental models represent how something works. Scripts provide sequences of actions relevant to a particular situation. Heuristics allow humans to react with a fairly good probability of success based upon their prior experience. Humans tend to generalize prior experiences to unknown or similar situations, often resulting in less than optimal performance.

Scholtz et al. [82, 83] categorize the methods to measure situational awareness commonly used in the field as follows:

1. **Performance-based** methods consider the action taken by the operator and determine the appropriateness or correctness of these actions.

2. **Knowledge-based** measures isolate components of situational awareness and assess them separately. Collecting this information is potentially intrusive into the operator’s task and in general is not used in actual tasks but is assessed in simulated environments.

3. **Verbalization**, such as think-aloud and talk-aloud methods, provide insight about the importance of information and the process that the operator uses. These methods are potentially intrusive to the operator’s task.

4. **Subjective measures** entered directly by participants, who use a linear scale and assign a numerical value to their situation awareness.

Both Verbalization and Knowledge-based methods can identify declarative knowledge better than procedural knowledge. Performance-based methods give better information about procedural knowledge.

An overview of commonly used situational awareness measurement techniques are provided by Yanco and Drury [100, 101]. In *Situational Awareness Rating Technique (SART)* [94], operators rate on a 10-dimensional bipolar scale the degree to which they perceive demand on the opera-
tor’s attention, supply of attention and understanding the situation. Another subjective technique
*Crew Awareness Rating Scale (CARS)* [58], is a generic 8-part questionnaire addressing both the
mental content and mental processing of situational awareness with respect to the three levels dis-
cussed earlier and an additional fourth level, *integration*, which requires synthesis of perception,
comprehension and projection with one’s courses of action. *Mini Sitreps* [59] is an implicit per-
formance measure based on short situation reports to provide an objective measure of the match
between the subject’s understanding of the situation and the actual situation at that point in time.
*Situation Awareness Global Assessment Technique (SAGAT)* [27] and *Real-time Probes* are explicit-
performance techniques employing periodic, randomly-timed freezes in a simulation scenario during
which the operator is asked a series of questions about the situation. In addition, as a subjective
measure, the NASA Task Load Index (NASA-TLX) is a widely used questionnaire where operators
rate the task load.

Situational awareness is used as a main metric in various studies. Humphrey et al. [44] 
evaluate a teleoperation type interface where the principle camera view is surrounded by a “halo”
showing the relative positions of other robots. situational awareness is measured using 3D SART, a
variant of SART. Scholtz et al. [82, 83] introduce an evaluation methodology for user interfaces for
supervisory roles in human-robot interaction for the on-road driving domain, where the operator
situational awareness has been measured using SAGAT.

Although situational awareness and performance are related, they are not directly corre-
lated. An operator can have perfect situational awareness and still perform inappropriate actions
due to sub-optimal decision making. Therefore, situational awareness is generally used with other
metrics to evaluate the quality of an HRI. Olsen and Goodrich [67] introduce some of these metrics,
 focusing on human supervision or control of robots. Their work considers HRI schemes where the
human operator supervises each robot platform sequentially and in turn the robots execute the
operator’s commands with some degree of autonomy. Six interrelated metrics are discussed, which
the authors deem as a guide for HRI design:

1. **Task Effectiveness** ($TE$) is a domain-specific measure of how well a human-robot team accomplishes some task. For multi-robot exploration, $TE$ is usually based on speed of performance, number of errors, damage caused in the environment and overall coverage of the area. It is not uncommon to use a combination of these metrics.

2. **Neglect Tolerance** ($NT$) is a metric representing the autonomy of a robot with respect to some task. In general, it is a measure of how its current task effectiveness declines over time, when neglected by the user. Defining the exact relationship is not always possible in practice. The authors hypothesize that there is a characteristic *neglect curve* that defines this relationship and by establishing an effectiveness threshold, $NT$ can be expressed in terms of time. In other words, $NT$ can be expressed as *neglect time*, which is the time between the last interaction with the operator and the time the neglect curve intersects with the effectiveness threshold\(^1\). Apart from task complexity, the user’s trust in the robots’ autonomous capabilities is also a factor that affects $NT$.

3. **Interaction Effort** ($IE$) is an abstract concept representing the overall effort the user shows to interact with the system. It is primarily a function of a human-robot interface and reducing this effort is the key to improving the HRI. Since most of $IE$’s components occur in the user’s mind, it is hard to measure.

4. **Robot Attention Demand** ($RAD$) is a unitless quantity that represents the fraction of a human’s time that is consumed by interacting with a robot, which is formulated as follows:

   $$RAD = IE/(IE + NT)$$

5. **Free Time** ($FT$) is the fraction of task time that the user does not need to pay attention to

\(^1\)In literature, the terms Neglect Tolerance and Neglect Time are often used interchangeably.
the robot. $FT$ is related to $RAD$ and formulated as follows:

$$FT = 1.0 - RAD$$

6. *Fan Out (FO)* is the number of robots that a human operator is capable of operating simultaneously and is formulated as follows:

$$FO = \frac{1}{RAD} = \frac{(IE + NT)}{IE}$$

This work shows that it is important to increase autonomy, which will then increase $NT$, solely a measure of the robot’s autonomous capabilities, and will decrease $IE$, which is the measure of interaction design effectiveness.

Although useful as a guide, $FO$, $FT$ and $RAD$ depend on $IE$, therefore cannot be directly observed. Olsen and Wood [68] explain the parameters that effect $FO$ and how they can be measured in different tasks and environments and propose the following formula:

$$FO = AT/IT$$

where $AT$ is the *activity time* or the time that a robot is effectively active after receiving a command and $IT$ is the *interaction time* or the time that it takes for an operator to interact with a robot. The rationale for this equation is summarized as follows. The primary interest in determining $FO$ is actually neglect time ($NT$) which is the time that the operator can neglect a robot and focus his attention elsewhere. However it is hard to measure this quantity. Consider a case where the operator gives a command for a robot to move from one location to another. If he does not trust the robot’s odometry and watches the robot, the robot is not considered to be neglected although the robot will move without intervention from the user. It is hard to capture this mental effort during
an experiment. Therefore $NT$ is expressed in terms of $AT$ in the formula, which is specifically the time that passes from receiving a command until the robot receives a new command, stops or completes its goal. The relationship between $AT$ and $NT$ is given by the formula:

$$AT = O \times IT + NT$$

where $O$ is the overlap or the percentage of interaction time when the robot is also active. For example, $O = 1.0$ for driving a car, since a car is almost always moving when the driver is steering it and $O = 0.0$ in the case of a manufacturing robot, since it is not active during setup but it will run for days or months after the setup.

The $IT$ part of the fan-out equation misleadingly seems easy to measure (such as measuring how long the user spends on one robot), however $IT$ is dependent on several factors, namely robot monitoring and selection, context switching, problem solving and command expression, which impact fan-out drastically. Since most of these occur in the user’s mind, they are difficult to measure directly.

The metrics examined so far applied to HRI systems known as “independent systems”, where the operator’s attention to a particular robot only affects that robot’s performance. According to Crandall [14], the robot effectiveness of an independent system can be depicted as in Figure 2.1. The underlying assumption here is that human input always increases robot effectiveness. In other words, robot effectiveness increases with increasing $IT$ and decreases with increasing $NT$.

For systems that require teamwork, especially environments where tasks require close coordination, such as box-pushing task, this model does not hold. Such systems are referred to as “dependent systems” and as Wang [96] describes, in a dependent system, when the operator is interacting with a robot, this interaction directly affects the performance of other relevant robots which are in close coordination with that robot. In fact, the performance of tasks such as box pushing cannot be improved unless each robot gets a fair amount of attention from the operator.
For these systems, the effects of $IT$ and $NT$ are as shown in Figure 2.2. This figure shows the overall effectiveness from the perspective of a single robot, in a two-robot dependent system. During the $NT$ for robot 1, the operator controls the relevant robot which indirectly effects the task in a positive manner. The occupied time, $OT$, represents the time spent in controlling the relevant robot and the free time, $FT$, is divided into two components; first, $FT$ represents the time spent off-task establishing team cooperation and the second represents the time spent off-task after team cooperation is established.

In the experiment I describe in Chapter 4, I will revisit these metrics as means to evaluate performance and situational awareness of the operators in dynamic environments. Specifically, I use a fixed number of robots in my experiments, therefore $FO$ is constant, despite the fact that $FO$
is a commonly used metric to measure HRI performance. The rationale for this lies in one of the goals of my work, which is to examine the factors that may potentially affect operator situational awareness and performance. When considering performance for an entire mission duration, \( FO \) is a suitable metric, but we are more interested in capturing specific situations that occur during a mission, that have an impact on operator situational awareness and performance. Hence, the metrics used in my studies are focused on indirect measures of situational awareness.

I use a combination of performance-based and subjective rating methods since knowledge-based methods like SAGAT and SART and verbalization methods are known to have an effect on performance, which we are interested in measuring. This approach will prevent me from examining each of the perception, comprehension and projection levels of operators’ situational awareness separately. This restriction is balanced by the way data is collected in my studies; each decision made by the operators during a mission is compared with a benchmark solution therefore yielding approximate performance scores. These performance scores, while not replacing accurate measurements of situational awareness, do provide some insight into all of these levels, since without good comprehension and projection, it is not possible to perform well in a consistent manner.

To measure performance, we use \( TE \), \( IT \) and/or \( FT \) depending on the particular analysis performed. It is important to note that the interaction studied in this work are different than interactions examined by Crandall and Wang. The operator’s role in my studies is specifically restricted to planning and task scheduling. In order to perform this role, operators give commands to the robots sequentially as in Crandall’s work, however in task scheduling context, some of the assignments have a direct effect on the performance of other robots, therefore the environment is dependent for some subset of tasks as Wang proposed. Therefore, during my analysis, modified versions of these metrics or additional ones are introduced.
2.2 Multi-Robot Task Allocation (MRTA)

One approach to SOMR control is to model it as a planning problem and decide a joint action for the whole team in order to reach the goal states. Although this approach considers all aspects of the problem, due to the dynamic nature and uncertainty of physical environments, it is computationally intensive. Task allocation is a key step in a multiagent planning problem [19], approaches to which are summarized in this section.

Task and resource allocation problems have been investigated rigorously in the MAS and MRS literature to achieve efficient coordination among team members. The approaches to task and resource allocation can be crudely categorized into centralized and distributed approaches. Centralized solutions usually require a certain level of reliable communication between the agents, but have the advantage of collecting all the information prior to reasoning and also require a relatively small amount of message-passing between the agents and the central solver. These methods work when the number of agents is limited, but they do not scale well. In addition, in the event of a possible failure of the central solver, the whole system collapses. Therefore they are not robust enough for dynamic and noisy environments. To address the robustness and scalability issues, which are critical for multiagent and multi-robot domains, distributed approaches are recommended and employed widely within these fields.

2.2.1 Market-based Approaches

Market-based approaches have been widely used in multi-robot task allocation to optimize resource usage, communication and task completion time [31]. The application domains vary from loosely coupled tasks, such as exploration [31, 103], to tightly coupled tasks [48, 31], such as box pushing which requires close coordination. The research in this area diverges, focusing on areas such as allocation protocols and mechanisms [90, 6, 103, 51, 53], bidding and clearing strategies [95], reallocation [34] and commitment strategies [57].
Wellman [98] describes market awareness as agents being able to interact using market mechanisms. Wellman states that every decision in a MAS is about resource allocation, where the term “resource” is used in a broad sense ranging from physical materials like memory, wires, or airplanes to more abstract activities like time, space, attention or expertise. The allocation of these resources is handled by multiagent interaction protocols or mechanisms. Market-based mechanisms base the allocation or exchange of these resources on a numerical value: price. Once agents’ intentions are expressed as price, the interactions take place in resource space or market space: market space = goods space × mechanism space. This perspective reduces communication significantly and supports distributed architectures.

Market-based approaches rely on local information or self-interest of the agents to solve large-scale complex problems. As Dias and Stenz state [22], centralized approaches can be computationally intractable, brittle and unresponsive to change; and distributed approaches can result in sub-optimal allocations from an individual’s perspective. Free market-based systems, on the other hand, overcome sub-optimal decisions by relying on individuals’ self-interest to effect the overall efficiency. The most common instantiations of market-based approaches in MAS and MRS are auctions, where resources and tasks are offered to all agents and each agent responds with its preference information for an item through a “bid”. In MRS, commodities distributed with auctions are not restricted to resources and tasks only. Frías-Martínez et al. [29] used auctions to distribute predefined roles to a team of robots in a game of robotic soccer, in order to deal with the highly dynamic nature of a soccer game; and roles for defensive and offensive positions were distributed in real-time.

One of the earliest applications of market-based approaches in multi-robot coordination is the exploration problem. One such application is by Zlot et al. [103] who describe their approach as a single-item, first-price auction mechanism to distribute the areas to be explored by means of goal points, which are points selected either randomly or based on the range of robots’ sensors. In this
scheme, the robots construct a path (list of goal points) and start running auctions in an attempt to outsource the points to another robot at a reduced cost. Since all the points are auctioned, the resulting plan is a good solution in terms of reduced cost for the team. The authors state that a globally good result emerges from the self-interested behavior of individual agents.

Single-item auctions are useful in the distribution of resources and tasks when they are not dependent on one another, however in most realistic multi-robot applications, this is not the case. It is common to have strong dependencies between tasks, which leads to sub-optimal allocations when agents acquire only some of the items they require to complete a task. Combinatorial auctions approach this problem by auctioning items in bundles.

A sub-optimal allocation example using single-item auctions in dependent task environments is presented by Berhault et al. [6]. As can be seen in Figure 2.3, one example is an exploration domain in a grid world where there are four tasks $G_1, G_2, G_3, G_4$ and two robots $R_1, R_2$. $R_1$ is in between $G_1$ and $G_3$, closest to $G_3$, and $R_2$ is in between $G_2$ and $G_4$, closest to $G_4$. In addition, $(G_1, G_2)$ and $(G_3, G_4)$ pairs are closer to each other. Although the optimal solution would be to assign $G_3$ and $G_4$ to $R_1$ and $G_1$ and $G_2$ to $R_2$, if these tasks were to be auctioned separately, $R_1$ would be assigned $G_3$ and $R_2$ would be assigned $G_4$.

![Figure 2.3: Sub-optimal single-item auction allocation [6].](image)

Although the idea is appealing, combinatorial auctions are computationally far more complex than single-item auctions. The two main problems with combinatorial auctions are auction design and winner determination. Although winner determination is NP-complete, there are approximation algorithms which perform well for small groups of robots.
Koenig et al. [51] compare Sequential Single-Item Auctions (SSI) with combinatorial auctions and Parallel Single-Item Auctions (PSI) auctions in a theoretical framework. In general, the SSI algorithm works as follows: in an environment where multiple robots bid on multiple tasks, the robots bid on every task (which are initially unallocated), based on some criteria like distance in an exploration setting. Each robot then bids its cost for adding the new task to its already-won tasks. This process is repeated until all tasks are allocated. The main difference between PSI and SSI is that SSI runs for multiple rounds, therefore has a better advantage of being assigned tasks which are closer in terms of bid criteria. It is also better in terms of computational time and message complexity than combinatorial auctions, which find the optimum allocation but are NP-hard. Empirical evaluations of PSI and SSI auctions are presented in our earlier works, in Schneider et al. [79, 80], and the results of these works support that SSI auctions perform well in solving real-time MRTA problems compared to other single-item auctions such as PSI. The testbed and scenario maps used in [79] include the ones that are used in the user studies described in Chapters 3 and 4.

In Chapter 4 an automated task scheduling tool is introduced. When discussing the implementation of this tool we’ll revisit auction mechanisms, particularly SSI.

2.2.2 Constraint Programming (CP) Techniques

A widely used method to describe complex problems is to describe them as a Constraint Satisfaction Problem (CSP) or Constraint Optimization Problem (COP). Due to their simple definitions and applicability in a large domain of problems, these approaches draw considerable attention from the MAS community. Essentially, a CSP is a problem to search for variable values that satisfy a set of constraints and a COP is a search for values that optimize a set of constraints. In the context of multi-robot task allocation, one model can assign tasks as the set of variables, robots as the set of acceptable values of these variables, and a set of constraints which restricts some combinations of assignments of robots to tasks. This is also a helpful abstraction to model complex tasks. Once
Modeled, methods to solve problems of this type can be classified as systematic search, consistency techniques, constraint propagation, stochastic and heuristic algorithms [4]. The MAS community, by its nature, is mainly interested in distributed versions of existing algorithms [13, 47, 55]. Some mixed-initiative planning and scheduling problems in complex task domains have been framed as CSPs by Smith [91, 92].

The classical description of CSPs as described by Yokoo and Hirayama [102] is as follows:

**Constraint Satisfaction Problem (CSP):** Formally, a CSP contains a set of variables:

\[(x_1, x_2, \ldots, x_n),\] where values are taken from discrete domains \(D_1, D_2, \ldots, D_n\), respectively.

A solution is then defined as a set of values that satisfy a set of constraints, represented as predicates, \(p_k(x_{k1}, \ldots, x_{kj})\), defined on the Cartesian product \(D_{k1} \times D_{k2} \times \ldots \times D_{kj}\).

Since CSPs are NP-complete in general, a trial-and-error exploration of the solution space is required. This is usually done through search and consistency algorithms. As covered in Rossi et al. [75], Dechter [20] and Yokoo and Hirayama [102] these algorithms fall into three fundamental classes:

- **Backtracking** is a commonly used systematic search algorithm, where variables are assigned values from a set that are consistent with the values of previously assigned variables. When a variable cannot be assigned a valid value, an alternative value to the previously assigned variable is sought and this step is called backtracking. Although, backtracking is a simple depth-first search algorithm, its efficiency depends on heuristics about how this search should be performed.

- **Iterative improvement algorithms** are commonly used local search techniques and work similarly to hill-climbing. An iterative improvement algorithm assigns values to variables according to the min-conflict heuristic. Each variable starts with a tentative initial value, a flawed solution; and, in iterative steps, the values of variables with the maximum number of
conflicts are reassigned to minimize the global conflict. The difference between backtracking and iterative improvement algorithms is that in the latter, all of the values of variables are changed in each iteration. As in all hill-climbing algorithms, getting stuck at a local minima is a problem.

- **Consistency algorithms** or **constraint propagation algorithms** are preprocessing algorithms that reduce futile backtracking by removing domain elements from variables that cannot be a part of a solution. In general, this class of algorithms makes use of the notion of *k*-consistency. A set of constraints, size \( k-1 \), are called \( k \)-consistent, iff the addition of any other constraint to this set would preserve consistency among them. This approach suffers from the computational burden of finding consistent sets of constraints when \( k \) is large. Algorithms such as *AC-1*, *AC-3*, *AC-4* for arc-consistency and *PC-1*, *PC-3* for path-consistency are examples of commonly used \( k \)-consistency algorithms [20].

Distributed versions of CSPs, called Distributed Constraint Satisfaction Problems (DSCPs), are suitable for defining a large set of MAS problems, where agents are concerned with finding consistent actions for coordination. Jung et al. [46] define DSCP as CSPs where constraints and variables are distributed among multiple agents.

**DSCP:** Formally, a CSP can be modeled as a DSCP, where the set of variables are distributed among a set of \( m \) agents, \( A = \{ A_1, \ldots, A_m \} \) such that each agent \( A_j \) has one variable. In addition, constraints are distributed among these agents, where any constraint might belong to several agents. If a constraint belongs to multiple agents, it is called an **external constraint** and if it belongs to a single agent, it is referred as a **local constraint**.

Yokoo and Hirayama [102] present several DSCP algorithms based on search algorithms used for solving classical CSPs, with the assumption that each agent holds a single variable:

- **Asynchronous Backtracking:** This algorithm is a distributed version of backtracking. Agents
asynchronously assign values to their variables and broadcast these values to other agents. Upon receiving a value assignment, agents check to see if they can assign a value to their variable consistent with it, and send out a *no-good* message if they fail to do so. Conflict resolution is carried out according to a priority ordering among agents.

- **Asynchronous Weak Commitment Search**: This algorithm addresses the issue of bad value selection for a higher-priority agent in a backtracking algorithm. Instead of performing an exhaustive search to revise the bad decision, this algorithm drops the flawed solution and restarts the process. Also, the algorithm takes the benefit of the *min-conflict* heuristic to reduce the risk of making bad decisions.

- **Distributed Breakout Algorithm**: As in the classic CSP version, this algorithm defines a weight to each variable therefore keeping track of variables with a large number of conflicts. In this distributed version, agents broadcast the weight values of their local variables. Also, agents communicate with *ok?* or *improve* messages, indicating that the broadcast values are consistent or the solution needs to be improved, respectively.

Various other algorithms are proposed to solve DSCPs and Distributed Constraint Optimization Problem (DCOP)s to this date. Some commonly used ones are described in Meisels [60]. For DSCPs the Asynchronous Forward-Checking (AFC) family of algorithms are described, which improves upon Asynchronous Backtracking with an additional step in the value assignment. The agent performing a value assignment first seeks approval from other agents by broadcasting the *current partial assignment*. This helps other agents to remove inconsistent values from their domains for later steps. If at least one agent ends up with an empty domain, the agent performing the value assignment tries another value from its domain. For DCOPs, ADOPT, an asynchronous version of Branch and Bound algorithm is presented.

An earlier work by Bejar et al. [5] is an example of a MAS problem modeled as a DSCP. In this work, the authors model a wireless sensor network domain as a DSCP, to track multiple moving
objects and refer to it as \textit{SensorCSP}. In their work, a wireless sensor tracking system consists of a set of $n$ Doppler radar-based sensors $S = \{s_1, s_1, \ldots, s_n\}$ that are required to track the position of a set of $m$ mobile objects $T = \{t_1, t_2, \ldots, t_m\}$.

Modi et al. [61] defines 3 properties to classify DSCPs in MAS task execution environment. According to this classification, if the constraints in the problem don’t change with time, then the problem is referred to as static, otherwise dynamic. If operations to complete a task don’t conflict with each other, in other words the operations can be carried out in parallel, then the task is said to be \textit{strongly conflict free}. If there exists a task execution scheme, where multiple tasks can be completed in parallel, then it is said to be \textit{weakly conflict free}. The last property refers to the ambiguity when an operation is carried out. The problem is ambiguous if the operation may be used to complete more than one task.

In order to solve problems in dynamic environments, Modi et al. [61] introduce \textit{Dynamic Distributed Constraint Satisfaction Problem (DyDSCP)}, also a variant of DSCP. The DyDSCP notion extends DSCP by allowing constraints belonging to an agent to change according to the varying environmental conditions. Therefore, each constraint $C$ is associated with a predicate $P$ forming a tuple $(C, P)$ and when $P$ is true, $C$ should be satisfied to find a solution. This provides an application of the DyDSCP approach to environments where agents do not know the constraints in advance, therefore they can observe the environment and make changes to their models of it. Also, due to changing conditions even if a solution is found, agents still need to keep track of $P$ values of constraints in order to find a new solution if necessary.

In general, the output and/or performance of any centralized or distributed CSP and COP algorithm depends on the order of search, heuristics and the structure of the constraint network, all of which are subject to change in a dynamic MRS. This poses several challenges from the perspective of interaction between a human operator and an automated planner agent. Despite this challenge, DSCP/DCOP is a popular model in MAS and MRS environments, therefore they are potential
candidates for SOMR interaction. I’ll further discuss the implementation of these techniques as automated planners, and potential impacts on the interaction between operators and planners, in the remainder of this chapter and in Chapter 5.

2.2.3 Algorithm Steerability

All MRTA approaches presented so far have been studied widely in various settings, mostly from the computational and communication complexity point of view. They also found use in SOMR control scenarios either as the distributed coordination mechanism among the robots or as underlying methods for decision support functionality to the human operators. As mentioned in Chapter 1, previous work suggests that such autonomy poses problems from an interaction point since black-box solutions generated by autonomous planners are discarded if they are not understood by the operators, which may leading to an overall performance degradation. From the HMRI standpoint, another property, steerability of MRTA algorithms is more important because it reflects algorithms’ controlability, which can be used to design an interaction scheme between the operator and the autonomous components.

As described by Baishya and Lewis [3], steerability can be used to classify teamwork algorithms based on the cognitive accessibility that they provide to a human operator. Their classification is as follows:

1. *Transparent*: This class of algorithms can easily be directed to produce a certain output.

2. *Translucent*: This class of algorithms have certain accessible cognitive parameters, however, their inner workings are not completely apparent to users.

3. *Opaque*: This class of algorithms are black-box solutions; their inner workings are too complex to form cognitive links between their input and output.

The example given for the transparent class of algorithms is path-finding algorithms which can
easily be manipulated via a mouse on a map by generating waypoints or by dragging certain parts of a formed path.

According to this classification, CSP and COP algorithms are candidate examples of the translucent class. In both problems, while the task environment can be expressed very naturally by humans using constraints, the cognitive link between the operator and the algorithm behavior is weak. The algorithms that solve CSPs terminate after finding a solution, while the algorithms used for solving COPs search for the best solution based on an objective function. However, they do not necessarily provide insight into how a search progressed or any means to compare different solutions. From an HRI perspective, CSP and COP solvers have some caveats that should be considered for interaction design:

- **No solution case**: CSPs have the potential to be turned into over-constrained problems, where no solution exists, in which case, complete algorithms will find no solution while incomplete ones will run without termination. Even if the algorithm terminates, the interaction problems get more complex requiring the relaxation of constraints.

- **Multiple solution case**: Both CSP and COP algorithms do not output information about the possible number of solutions. From a decision-making perspective, a situation with only a single possible solution and a situation with many solutions are vastly different, however judging by CSP and COP algorithms’ output, these two cases are indistinguishable.

- **Vast search space case**: According to the description by Meisels [60], the parameters, constraint density ($p_1$) and constraint tightness ($p_2$) are used for specifying the general structure of the constraint network; $p_1$ refers to the density of the edges (constraints) in a constraint graph, while $p_2$ specifies how restricted the constraints are (or size of the variables’ domain list that satisfy the constraints). The phenomenon mentioned is that when $p_1$ and $p_2$ are at a certain range, time to reach a solution peaks. This is observed when both parameters are not too high and not too low. This results in a constraint graph where the algorithms cannot
find a solution easily or if a solution does not exist, they cannot exhaust all possible variable assignments quickly. From the user’s perspective, this phenomenon may lead to long wait times, unacceptable in real-time missions.

Similarly, auction mechanisms, from an HRI standpoint, can be considered translucent. Since the information that is passed from robots to the auctioneer is only a single bid, which summarizes all decision factors that an individual robot takes into account, it is very difficult for a human operator to dissect this bid into its constituent factors and figure out the underlying reasoning behind the robot’s bid, especially in real time. However, the auction results can be manipulated via auction clearing rules and even bidding strategies of robots with the assumption of benevolence, which holds for teamwork. Although these manipulations may not be interpreted as a direct description of the desired solution as in the case of defining constraints, it could be argued that they certainly provide some degree of cognitive accessibility which puts auctions apart from opaque algorithms.

Regardless of this less descriptive property, auctions have several advantages over CSP and COP approaches. First and foremost, auctions are robust against single-point failures. Even during planning, failure of a robot to send its bid to the auctioneer can be recovered by ignoring the input of that robot or repeating the auction. Repeating an auction, if it is not a form of combinatorial auction, takes linear time to clear therefore makes this approach feasible and also superior to CSP and COP approaches. This is true even if the problem is of combinatorial nature, since SSI is shown to perform as well as combinatorial auctions in a variety of problems [51]. Lastly, auctions always return a solution even if it is partial, which makes them “anytime” algorithms. From the interaction perspective, this resolves the problem of relaxing and redefining a problem as in the case of CSP. It is important to note that COPs are also capable of returning a partial solution if not all constraints are satisfied, however they can by no means be considered as anytime algorithms, since solving COPs are computationally intensive.
In Chapter 5, I revisit these two categories of approaches and discuss how they can be steered based on the findings of studies conducted in Chapters 3 and 4.

2.3 Related Work

Using intelligent autonomous agents as teammates (either software or robotic platforms) has been studied in a variety of contexts. A thorough treatment of roles, applications and types of interaction between human operators and these agents can be found in [93]. Potential applications, interaction context, operator and UV roles have been investigated both for *Unmanned Aerial Vehicles (UAVs)* and *Unmanned Ground Vehicles (UGVs)* in various approaches and empirical studies. Some of the related work is outlined in this section.

For UGVs, *Urban Search and Rescue (USAR)* has been a significant challenge. Wang [96] investigated the effectiveness of Single-Operator Multi-Robot (SOMR) control in high-fidelity, realistically complex settings. This work focused on UGVs, and tasks were complex in terms of the number of required platforms and tight-coupling between robots, such as box pushing. Gao et al. [30] introduced multiple operators in the control loop and conducted experiments to observe the effects of team structure. Despite the anticipated extra amount of workload related to coordination, the potential for increasing team size makes this approach appealing. Experiments were conducted in the *USARSim*\(^2\) environment with two operators controlling 24 UGVs, either divided amongst them or forming a shared pool which operators accessed when they needed additional robots. In military settings, Carlin et al. [9], experimented on domains where UGVs were co-located with humans in the same environment to perform *room clearing*. A distinction was made between *task points*, i.e., locations in the room where the UGVs could potentially cover larger areas. Certain task points were determined to be more *critical* than others, such as corners.

At one end of the spectrum, some researchers investigated human-in-the-loop deployment

\(^2\)http://sourceforge.net/apps/mediawiki/usarsim/
of large-scale teams. One such work by Hardin and Goodrich [39], focused on the level of autonomy of UGV agents and its effects on team performance. Experiments were conducted on operators in control of 200 robotic searchers in a Wilderness Search and Rescue simulator. The level of autonomy for agents’ responsibilities was decided dynamically. In adaptive autonomy mode, agents were given exclusive control. In adjustable autonomy mode, the human operator was given control, and in mixed-initiative mode, the agents and operator worked in collaboration. Mixed-initiative control outperformed other controls during experiments, which was expected since human managers were better at focusing on likely search areas, while autonomous agents tended to utilize large groups of agents more effectively. Another work by Kolling et al. [52] proposes a mechanism to supervise a swarm of robots ranging in numbers from 50 to 200. The main mechanism proposed is based on selection of a subset of the swarm and placing beacons which modify the behavior of the nearby swarm in conjunction with a set of behaviors. Using this mechanism, operators are capable of forming coalitions using the selection tool and assign both low level behaviors such as stop, come, rendezvous, deploy etc. and more complex behaviors such as foraging. With the utilization of beacons, which attract or repel the nearby robots, operators can further direct the behavior of the swarm or in other words steer the underlying algorithms responsible for the execution of the particular behavior.

At the other end of the spectrum, similar to the scope of this dissertation, some researchers focus on supervisory control of multiple UVs at a tactical level. In this context, automated resource allocation to support decision making by human operators is studied. As stated by Clare et al. [12], the benefit of humans being in the control loop, when computers can do optimization much better, becomes apparent in making complex decisions such as prioritizing tasks, interpreting camera imagery, etc. Such decisions are examples of “knowledge-based reasoning” where humans are superior compared to computers. This is due to our superior improvisation, flexibility and inductive reasoning abilities. Also computer optimization algorithms are notoriously “brittle” in that, they
can only take into account those quantifiable variables identified in the design stages that were deemed to be critical.

In some of the most relevant and recent work, a number of problems in this domain are addressed. In [18, 66], Cummings and Nehme introduced a discrete event simulation approach to human-system modeling that includes a quantitative relationship between workload and performance. The authors’ model is based on a queuing system where the operator is conceptualized as a server that needs to address both endogenous and exogenous situations. The authors characterize operator strategies such as first-in first-out (FIFO) and highest attribute first (HAF), which are typically employed in high risk command and control settings. This view of human operators serving each platform in sequence is important in the choice of strategies employed by the operators. As part of the research questions addressed in this dissertation, similar analysis is done regarding operator strategies in Chapters 4 and 5.

One particular statement that the authors make in [18] regarding the relationship between mental workload and the “percent busy time” is important from the perspective of my work. Mental workload is fundamentally determined by the relationship between resource supply and task demand. Representing the concept of utilization, percent busy time is an effective proxy for measuring mental workload. Numerous studies examining the supervisory controller performance support that when tasked beyond 70% utilization, operators’ performance decline. This relationship between the mental workload and the percent busy time will be utilized in the analysis in Chapter 4.

In [17], Cummings and Bruni suggest that decision support systems should be used to collaborate with human operators, which typically have limited computation capability and are biased in decision making when faced with uncertainty during solving complex task allocation problems. The use of satisficing or accepting a good enough solution is key in dynamic, uncertain and high-risk settings. However, when used without considering human factors, automation-generated solutions
often cause more confusion, loss of situational awareness and automation-bias on the part of the human operator. This statement is also embraced in my work and describes the motivation behind my approach to the problem space considered in this thesis.

In the work of Cummings and Bruni [17], various decision support approaches were tested to observe if they could improve operator performance in resource allocation problems. The problem domain of interest was missile strike planning where the operators plan a list of resource task pairings which correspond to missile platforms and targets. Experiments were carried out based on the information type classification: hard-constraints, probabilistic information and optimization information. The experiments were conducted using three different collaborative interfaces: human-centric (manual assignment), balanced (manual assignment and auto-complete functionality with weight heuristic) and automation-centric (auto-complete with limited heuristics) which implemented these information types. The results show that the manual assignment yields the highest performance and the worst workload whereas the automation yields the worst performance with best workload results. The balanced approach yields in between results. Approximate mission planning time for the experiments is measured in days. The authors rightfully argue that the operator behavior and strategy most probably will be different where time stress is increased in missions where planning time is expected to be in seconds or minutes.

In an extension of this work, resolving scheduling conflicts for time-critical UAV targeting missions was studied in [16]. Experiments were conducted comparing the support given for local and global scheduling conflicts. A comparison of human-automation collaboration modes was presented from a command-and-control standpoint, ranging in degrees from automation-centric to human-centric.

In [10], problems related to the presence of unknown variables and possibly inaccurate information, leading to reduced performance level of automated planning and scheduling algorithms, were examined. The results of this work suggest that task-based interfaces may reduce operator
cognitive overload and increase the efficiency of SOMR control interfaces for automated planning and scheduling.

Steering or guiding algorithms in complex task environments were also investigated. As Clare et al. [11] state, algorithms used for task allocation and path planning generally have an objective function which is set at design time. This approach doesn’t always work in dynamic environments, therefore the authors suggest guidance of these algorithms using a weighted objective function which can be manipulated at run time. Experiments were conducted in a multi UV command and control scenario, where the operators act as supervisors for a heterogeneous team of UAVs with the goal to search, identify and destroy targets. The operators were allowed to prioritize factors such as area-coverage, search/loiter tasks, target tracking, hostile destruction and fuel efficiency. The results showed that subjective performance and confidence ratings of the operators were highest when using the dynamic objective function. I use a similar approach in this work as described in Chapter 4.

Clare et al. [12] then further investigated interaction schemes where the operators’ main role is to approve auto-generated task schedules rather than individually assign tasks to UAVs, generate search tasks and identify targets. In this scheme, the work investigates the impact of fixed-time replanning intervals (30, 45 and 120 seconds) on the operator workload and performance in the same command and control scenario as [11].

### 2.4 Thesis Scope

In this last section of this chapter, I define the scope of this dissertation in three parts. First, the task allocation problem that the human operators face will be described from the computational perspective. Next, the descriptions of some terminology used throughout this dissertation is clarified. Finally, the scope of this dissertation is summarized and the main difference between this work and previous work with similar scope is highlighted.
An earlier model for task allocation in MRS settings, as described in [76], is called the Generalized Assignment Problem (GAP) which seeks to maximize a reward value by assigning as many robots as possible to a set of independent tasks, and is known to be NP-Complete. The main restriction of GAP is that it doesn’t encapsulate inter-dependencies among tasks and the coalitions required to execute MR tasks. Scerri et al., in the same work [76], remedy these shortcomings by introducing the Extended Generalized Assignment Problem (E-GAP), that allows additional constraints to be specified among tasks therefore enabling coalitions to be formed to execute MR tasks. In addition, in this work, their approach seeks to maximize not just the reward that comes from a static allocation instance but also from a sequence of allocations, therefore robots can be used repeatedly during the course of the mission. The E-GAP is also known to be an NP-Complete problem.

The task allocation problem that is considered in this dissertation is very similar to the E-GAP with two differences. The major distinction is the inter-dependencies among the tasks in E-GAP considers only if tasks need to be executed at the same time, while in this dissertation, due to spatial restrictions, some tasks may depend on completion of other tasks, imposing an implicit ordering of execution. The second distinction is that the E-GAP is defined as a problem for maximizing reward, therefore the algorithms will seek allocations that may not encapsulate assignments for all of the tasks. Although theoretically there is no difference in the presented work and the E-GAP, in my analysis I do not cover these situations with partial assignment of tasks.

Some of the related terminology is used ambiguously or in a relaxed form in the literature, therefore, here I provide clarification as to how these terms are used in this dissertation. In the AI planning field, “planning”, in general terms, refers to the deliberation process before acting. Resource allocation and scheduling, which is the focus of this work, is defined as a sub-field of planning [65]. Therefore the term “plan”, as used in this dissertation, refers more specifically to the order of execution of tasks and the assignment of specific platforms that will execute them.
I’ll use “plan” and “schedule” interchangeably throughout the dissertation, although a schedule signifies a plan in which expected execution times of tasks are concrete.

Also in the literature, algorithms and systems capable of solving task allocation problems in HMRI are variously referred to as “mission planners”, “autonomous planners”, “autonomous task planners” and “autonomous schedulers”, depending on the capabilities of the system and the context. I will use the term “autonomous planner”, as used by Clare et al. [12], to refer to the task allocation algorithm used in the system described in this dissertation.

Finally, the use of intelligent agents as teammates may refer to a variety of things, depending on the LOA and the role of the agent. In this dissertation, I use this phrase to refer to an autonomous planner which interacts with the human operator in a decision support role.

The aim of this dissertation, as stated in Chapter 1, is to study and improve human supervision of multi-robot teams through usage of human-guided automated planners as decision support tools to help reduce operator situational awareness. The domains considered in this work are complex and dynamic task environments. As covered in Chapter 2.3, there is existing work that focuses on dynamic domains, however, a limited number of these consider complex task allocation environments. Among the ones that do, there are different interpretations of the term “complex”. Some of this previous and recent work focuses on real-world problem environments, such as UAV mission planning, that has the same goal as this dissertation.

The main contributions of the work described in this dissertation, that differs from existing work, lies in the generality of the approach and the scope of application. While existing work focuses more on defined and concrete problem domains, the work described in this dissertation is more concerned with building a generalized formal approach. Without assuming detailed knowledge about the problem domain, I focus on formalizing the problem based on the common properties of potential mission environments that affect the decision-making and strategies of human operators. These common properties include spatial characteristics of the environment, task properties, plat-
form or robot capabilities and finally the distribution of tasks and robots in the environment. In
Chapter 3, these properties are modeled and analyzed in static problem instances, then in Chap-
ter 4 the model is extended to real-time dynamic missions and finally, in Chapter 5, I address and
discuss the question of how the behavior of the autonomous planner can be modified to improve
interaction with the human operator.
Chapter 3

MRTA Complexity Analysis in Static Environments

This chapter looks into the factors that make a task assignment problem cognitively and computationally complex. First, in Chapter 3.1, two taxonomies that are commonly accepted in the MAS community are presented. Then, in Chapter 3.2, an overview of the task structures that are commonly studied in MRS research is provided, in order to establish a general understanding of the expectations from MRS systems and complexities that arise from the properties of individual tasks. Based on these taxonomies and individual task properties, our analysis regarding the task landscape, is described. Next, in Chapter 3.3, a key idea in this dissertation, a model which we refer to as the TAG will be introduced. A TAG is essentially a graph-based model that encapsulates the spatial distribution of tasks and platforms in the physical environment. The purpose of this model, as explained in earlier chapters, is to identify the spatial and task specific factors that affect the mental workload and performance of human operators when dealing with task scheduling problems. A software tool, TCAT, is developed (see Chapter 3.4) and factors derived from the TAG are verified through a user study conducted with TCAT, in static problem environments, which is discussed in Chapter 3.5.
3.1 Overview of Task Taxonomies

The earliest widely cited taxonomy for multi-robot task allocation was introduced by Gerkey and Matarić [32]. The motivation of their work was to fill the gaps and unify the theoretical definitions of the multi-robot coordination field, enabling researchers to compare their results with those of others in the field, in terms of algorithmic efficiency and complexity. For this purpose, the authors introduced a taxonomy to define the types of problems that were faced in the field. Similarities with problems in other areas are discussed wherever possible. According to the authors, most problems can be categorized into a classification of task environment composed of three dimensions:

- **Single-task vs. Multi-task robots (ST,MT)**: Defines the number of tasks that robots can handle simultaneously.

- **Single-robot vs. Multi-robot tasks (SR,MR)**: Defines the resource requirements for task completion.

- **Instantaneous vs. Time-extended assignment (IA,TA)**: Defines the type of information that is available about the arrival rate or nature of future tasks.

From the computational perspective, this taxonomy provides a mapping of complexity class to a particular problem. As an example, the ST-MR-IA class of problem involves tasks that require combined efforts of a group, which makes these types of problems similar to the Set Partitioning Problem [62].

Landén et al. [54] extended Gerkey and Matarić’s work, by introducing four new dimensions:

- **Unrelated vs. Interrelated utilities (UU,IU)**: Techniques based on the concept of utility maximization are common in multi-robot coordination. This dimension signifies the difference between unrelated vs. related utilities, which is the case if the expected utility of task $t_i$ depends on acquiring task $t_j$ for some $i$ and $j$ in the task list. Domains with interrelated utilities are harder in terms of assignment.
• **Independent vs. Constrained tasks (IT,CT):** This dimension separates domains with independent tasks, which can be executed in any order or at any time, versus constrained tasks, which are required to be executed in an order, in parallel and/or within a time frame.

• **External vs. Internal Allocation View (EV,IV):** This dimension refers to the aspect of who is allocating tasks. In external allocation view, task allocation is carried out by a particular platform, while in internal allocation view, agents decide whether or not to perform a task. This is related to the method of allocation being centralized or distributed. In EV, using centralized techniques, all the information is assumed to be available to the platform making the allocation, while in IV, this is not necessarily the case.

• **Static vs. Dynamic Allocation Environment (SA,DA):** This dimension differentiates between static domains, where the constraints do not change during execution, and dynamic domains, where changes in the constraints may force reallocation of tasks.

Taxonomies discussed in this section allow researchers to identify the problem class and available methods to solve it and also to compare results with others.

### 3.2 Task structure in Multi-Robot Systems

Although the task taxonomies described in the previous section are very useful to classify task assignment problems as a whole, they do not distinguish the properties that define the environments and the properties of individual tasks. As an example, Gerkey and Matarić’s (ST,MT) and (IA,TA) dimensions and Landén et al.’s (UU,IU), (EV,IV) and (SA,DA) dimensions represent factors that have nothing to do with the task properties but encapsulate all other factors such as capabilities of the robots and environment characteristics that change the task assignment problem in terms of computational complexity. In addition to the dimensions described in these seminal works, we believe that it is important to distinguish properties of individual tasks that change the complexity
of an assignment problem, especially from the human operator’s perspective. The task properties that we adopt in this thesis, in order to address this issue is as follows:

- **Required platforms – multi-robot (MR) vs. single-robot (SR):** This property is the same one described in Gerkey and Matarić’s taxonomy [32] and applied to define a single task. In other words, we use the label MR as a property of a single task as opposed to the property of the problem instance and by doing so, we analyze the effects of the amount of MR tasks in a problem instance on the human operator’s performance and mental workload. An important note about this property is that from the human operator’s perspective, there is a distinction between an MR task requiring a fixed number of robots and one that would be executed faster with multiple robots, but the exact number of required platforms is not specified. We hypothesize that the latter would require additional mental workload on the operator, as it would require solving an additional *Coalition Formation* problem as commonly referred to in the MAS field [87].

- **Frequency – achievement (AC) vs. maintenance (MN):** [99] This property distinguishes between a task with a goal to reach a state and a task with a goal to stay in a state. Unlike an achievement task, a maintenance task such as patrolling a route would require periodic checks and updates leading to additional mental workload.

- **Constraints – independent (ID) vs. constrained (CN):** [54] This property represents the constraints imposed on the task by users or by the environment. A constraint on a task can be one of the following:
  - Temporal, such as deadlines.
  - Explicit dependencies on other tasks such as \( t_i \{ \text{before, during, after, ...} \} t_j \), for example, when a part needs to be picked up \( t_i \) before it is assembled to a frame \( t_j \).
  - Implicit dependencies imposed by the environment such as spatial constraints (e.g.,
access to the room where a fire broke out \((t_i)\) is blocked by debris which needs to be cleared \((t_j))\).

While all of the above categories are temporal in the classical use of the term, we use the word *temporal* to refer strictly to absolute time constraints (e.g., “in 5 minutes”, “by 3:00PM”, etc.) and explicit dependencies to refer to specified constraints on tasks in relation to one another. Both temporal and explicit constraints are considered to be mission parameters set by the operator or a higher authority. Implicit dependencies arise in situations where a task is spread over a geographic area and may temporarily block access to certain regions of the environment or resources. Examples of these types of dependencies may be given as fire or debris removal in a USAR scenario.

From an HRI perspective, interaction between the system and the human operator to set, modify and display these constraint types are different from each other in terms of cognitive accessibility, therefore require different approaches. Implicit constraints are spatial in nature, therefore easier to display in a mission map. Whereas explicit constraints are ordered and temporal constraints are numerical, therefore require additional components in the interface, such as visual tools similar to Gantt charts. The main difference between explicit and temporal constraint types is that from a cognitive perspective, it is easier to think and reason about precedence or ordering than it is to think and reason using numerical values.

- *Granularity – composite (CP) vs. atomic (AT)*: We refer to tasks that contain a fixed procedure for execution as atomic, such as a single sensor-sweep task. Composite tasks, as the name suggests, are composed of a set of atomic tasks, which may or may not be dependent on each other. An example of a composite task is patrolling an area, which consists of a set of points the robot needs to reach and then perform a 360° sensor-sweep at each of these points. In order to execute a composite task, further decomposition is required. In the example task of area patrolling, the operator has to think about which points to visit and the order to be
visited, if there is no pre-scripted plan for executing the task. Furthermore, in case the composite task is composed of independent atomic tasks, it is possible to assign multiple robots to divide the task into sub-problems, making it inherently an MR task without a fixed number of required platforms. As described in the required platforms property, this property also requires the operator to consider forming coalitions therefore increasing the mental workload.

A subset of previous MRS research are selected and their task environments are classified according to these four properties in Appendix A.

3.3 Task Assignment Graph (TAG)

In the work presented here, we consider structured indoor environments with multiple UGVs supervised by a single human operator. In indoor environments, UGV movements are restricted by physical obstructions such as walls. These restrictions become more of a hindrance if the environment is crowded by other agents or in settings like USAR due to additional obstacles and/or hazardous areas. From the supervisor’s perspective, we predict that these restrictions will become a key issue in future applications of human multi-robot teams, along with handling tasks that require close coordination among multiple robots.

Our model focuses on two of the dimensions mentioned in Chapter 3.2: Multi-Robot (MR) vs. Single-Robot (SR) tasks for representing platform requirements, and Independent (IT) vs. Constrained (CT) tasks for representing dependencies among tasks. In the user study discussed here, we specifically focus on implicit dependencies. As described in Chapter 3.2, implicit constraints or dependencies arise in situations where a task is spread over a geographic area and may temporarily block access to certain resources or regions of the environment. Examples of these types of dependencies may be given as fire or debris removal in a USAR scenario.

Our initial approach for our model is based on the intuition that these spatial relationships play a key role in the mental demand required for devising a mission plan. I next describe our model
and then present the results of a user study, conducted to verify the model. In order to understand operators’ expectations and behaviors under ideal circumstances, and to compare a feasibly large set of scenarios, we isolated the task assignment problem from real-time execution for the study described here. During the user study, participants were asked only to devise an initial plan for a series of scenarios, without any further interaction during the robots’ execution of the tasks. The experiments described in Chapter 4 examine extended interaction between a human operator and robot team during task execution.

To articulate the range of scenarios described above, we have developed a graph-based data structure, which we refer to as a TAG. A TAG represents spatial relationships between the tasks and the robots. In an environment containing $m$ robots, $R = \{r_1, ..., r_m\}$, and $n$ tasks, $T = \{t_1, ..., t_n\}$, we define $TAG = (V, E)$ as a set of vertices $V = \{v_1, ..., v_n\}$, where each vertex, $v_i$, represents a task, $t_i$, in $T$, and a set of edges $E$. There exists an edge in $E$ between any two task vertices $v_i$ and $v_j$ iff tasks $t_i$ and $t_j$ are accessible from one another in the geographic map representing the robots’ physical task environment. Each task vertex $v_i$ contains a set of robots $Acc_i$ that can access task $t_i$ and a domain $Dom_i$ which consists of sets of all possible assignments for that task. Finally, the cardinality of each set in $Dom_i$ is defined by the number of robots, $req_i$, required to execute task $t_i$. Essentially, a TAG is a hybrid graph structure that combines a spatial connectivity graph and a constraint network. This level of abstraction allows us to focus on spatial relationships in our analysis, without paying further attention to other specifics of the mission domain or map.

Once a scenario is represented as a TAG, we further label each task vertex, $v_i$, as critical if at least one of the following conditions is satisfied:

(i.) Removing $v_i$ from the TAG results in multiple disconnected components;

(ii.) Task $t_i$, together with other tasks, is jointly responsible for robots reaching at least one other task in $T$; or

(iii.) Task $t_i$ belongs to a group of tasks which prevent a robot cluster (group of robots co-located
in the same part of the map and able to access the same set of tasks) from reaching some subset of $T$.

The difference between conditions (i.) and (ii.) is that in (ii.), there are a number of tasks that block access to a part of the map and only one of these tasks needs to be executed in order to gain access to that region. The condition (iii.) refers to tasks trapping robots to certain areas of the map, which cannot be executed by the trapped robots alone.

The rationale behind labeling task vertices in (i.) and (ii.) as critical is based on our assumption that these characteristics will induce additional mental workload on the operator’s planning process due to the constraints they impose on the task allocation problem. The last type of labeling was based on observations during our earlier pilot experiments, where a significant number of participants stated that they were “compelled to free a trapped robot” and they placed high priority on tasks that would achieve this.

Figures 3.1 and 3.2 show examples of two different scenarios. In Figure 3.1, tasks $t_4$ and $t_6$ are both labeled critical due to condition (i). Task $t_4$ requires 3 robots for execution and confines robot $r_1$ to the left side of the map, therefore it is also labeled critical due to condition (iii). In Figure 3.2, tasks $t_2$ and $t_3$ together prevent access to $t_1$ and are labeled critical due to condition (ii).

We have defined a number of metrics to categorize the experimental scenarios that are presented to human subjects in our user studies.

To represent the SR-MR dimension of the taxonomy, we use the Average Platform Requirement (APR):

$$APR = \frac{\sum_{i=1}^{n} req_i}{n}$$

where $n$ is the total number of tasks to be assigned.
To represent the IT-CT dimension, we use the Critical Task Ratio (CTR):

\[ CTR = \frac{|V_{critical}|}{n} \]

where \( |V_{critical}| \) is the total number of critical tasks in the scenario.

### 3.4 Task Complexity Assessment Tool (TCAT) Test bed

The Task Complexity Assessment Tool (TCAT) [69] was used to present a set of scenarios to human subjects and evaluate users’ perceived mental workload and performance, with respect to each scenario (see Figure 3.3). The interface has two main components. The area on the left, the map component, displays a map of the robots’ environment, including the positions of the robots, the locations for performing the tasks, and the projected paths of robots as they are assigned to tasks. The area on the right shows the allocation component, which is manipulated by the human...
Figure 3.2: Depiction of a scenario where tasks $t_2$ and $t_3$ are critical based on condition (ii) (top). TAG corresponding to scenario above (bottom).

user, via point-and-click actions, to assign robots to tasks.

In the allocation component, users assigned robots to tasks by first selecting the robot(s), by a single mouse click, then selecting a task. Only the required number of robots were assigned, if the number of selected robots exceeded the required number of robots for that task. To deselect a robot, users had to click again on that particular robot icon. In order to deselect all the icons, users had to click on an empty area within the allocation component. The assignments could be removed by one of three ways:

1. Select a single assignment, represented by a line between a robot and a task, then hit delete key, which removed that assignment only,

2. Select a single robot then hit delete key, which removed all assignments of that robot, or

3. Select a single task then hit delete key, which removed all assignments made to that task.

One important detail regarding the treatment and usage of the TAGs, needs to be explained. For each scenario, the TCAT produces two copies of the TAG representing the initial situation of that scenario. One of the copies is used to keep track of changes in the scenario incurred by the assignments made by the users. For example, when a critical task is fully assigned, it is removed
from this TAG copy and the domain of the remaining task vertices are updated accordingly, in order to allow for additional assignments. The other TAG copy is kept unchanged and used in the analysis presented in Chapter 3.5.4, to categorize the scenarios into predefined complexity classes.

**Figure 3.3: TCAT Interface.**

Before moving on to the verification of the model, it is important to explain why our model is limited to structured environments. Our TAG model is meant to represent the environment from the human’s perspective at a high level. We believe this to be a reasonably valid assumption in structured environments based on the fact that:

1. Floor plans are widely-used area maps to represent structural environments and are generally understood by humans.

2. Various automated methods exist to generate topological maps using floor plans produce common features like junctions, rooms, openings and doorways, in the same manner.

Works of Dedeoglu et al. [21], Setalaphruk et al. [85] and Schaefer et al. [84] are examples of the second item. Dedeoglu et al. [21] present methods to generate autonomous real-time topological maps, which consist of features such as corridors, junctions, doorways, corners and rooms which
are represented as cul-de-sacs. Although this method does not require an *a priori* map (e.g., a floor plan), the experiments conducted in a structured environment successfully produce topological maps. Setalaphruk et al. [85] describes an algorithm based on Voronoi diagrams to convert a given floor plan to a topological map. Finally, Schaefer et al. [84] tackles a similar problem as Setalaphruk et al. and proposes an algorithm that parses architectural CAD-files and generates a topological map. In all of the above works, the topological map is generated by first adding a node to all center points of rooms, junction points and corners and then each node is connected to all its immediate neighbor nodes that are in the line of sight. We use the same topological map features during the design of the maps for our user study scenarios.

### 3.5 TCAT User Study

We conducted an experiment with human subjects to measure the degree of perceived difficulty of a task assignment problem instance in relation to the average $APR$ and $CTR$ values across a range of experimental scenarios.

Our experiment was designed to assess two hypotheses:

1. *an increase in APR leads to higher mental workload*

2. *an increase in CTR leads to higher mental workload*

Pilot studies, which we conducted earlier, suggested that there is an upper bound for $APR$ such that this hypothesis would hold. During the pilot study, for scenarios with very high $APR$ values, where most of the tasks required the majority of available robots, participants assigned all the robots to these tasks together—i.e., making the obvious assignment, which resulted in low mental workload. However, similar to the effect of high $APR$, in our pilot studies, very high $CTR$ values also suggested that lower mental workload occurred due to fewer options being available for solving these types of problems. Since most of the interesting cases lie in the region between the lowest and
highest extremes for these two parameter values, during the experiment presented here, we only used scenarios where the number of multi-robot tasks (vs. single-robot tasks) and the number of constrained tasks (vs. independent tasks) each comprised less than half the total number of tasks.

### 3.5.1 Experimental Setup

Our experimental scenarios are inspired by the RoboCup Rescue Simulation domain [50] where heterogeneous groups of agents attend to victims, fires and roadblocks in the aftermath of an earthquake in an urban environment. In our scenarios, a fixed number of robots (homogeneous and single-task (ST)) and tasks (single-robot (SR) and multi-robot (MR)) are scattered in an office-like environment.

The tasks can be any of the following: (i.) a *sensor-sweep* task, where a robot is expected to go to a specific location, gather and send sensor information (e.g., camera feed) back to a human operator; or response tasks resulting from two different events, namely *fires* and structural collapses creating *debris*, (ii). fire extinguishing and (iii.) debris removal. These latter two types of tasks may require multiple robots to execute, and they may block access to areas adjacent to those in which they appear. Sensor-sweep tasks can be executed by a single robot and do not block access to other tasks.

Three different types of environment maps were designed and tested for each category, in order to introduce some variance into the experiment. Note that these variations were not considered separate variables in our analysis for two reasons: first, the environment maps are represented in the TAG as connections between tasks and thus the differences are already accounted for in another experimental variable, *CTR*; and second, there are infinitely many map configurations, and we cannot make any inferences based on only three types, other than to say that our methodology is not limited to a single topology. Figure 3.4 shows the maps used in the experimental scenarios. Each map is designed to introduce a different topology. The map on left in the figure represents
a short corridor linked to a number of rooms, typical in indoor environments. The underlying topology can best be described as a connected sparse graph, where the degree of end nodes and portal nodes are fixed. The map in the middle represents a lobby area connecting directly to a number of rooms, with an underlying star topology. Lastly, the map on the right is composed of two connected ring topologies representing an entire floor, typical of hotels, offices and hospitals. During the generation of our experimental scenarios, as a rule, tasks that are potential obstructions are only placed in the map such that each task’s placement caused only one of the connections to be blocked.

Figure 3.4: The three types of environmental maps and corresponding topologies.

3.5.2 Procedure

Twenty-seven experimental scenarios were defined, with values: \( APR \in \{1.0, 1.5, 1.667\} \) and \( CTR \in \{0.0, 0.333, 0.5\} \). The values for \( APR \) come from scenarios where all tasks are SR tasks \( (APR = 1.0) \), 3 of the 6 tasks are MR tasks, requiring 2 robots \( (APR = 1.5) \) and 3 of the 6 tasks are MR tasks, with 1 of them requiring 3 robots and the others requiring 2 robots. Similarly, for \( CTR \), the values come from scenarios where there are no critical tasks \( (CTR = 0.0) \), 2 critical tasks \( (CTR = 0.333) \) and 3 critical tasks \( (CTR = 0.5) \) out of tasks. Scenarios are divided into four categories, based on their \( APR \) and \( CTR \) values. Each experimental scenario falls into one of the four categories, labeled Q1 - Q4, as shown in Table 3.1. Individual scenarios can be seen in Appendix B.

The user study followed a within-subject design, where all participants saw the same set of
scenarios in randomized order.

Each human subject was initially presented with two training scenarios and asked to perform a range of interface-related tasks covering all types of interactions. Then they used the interface until they became comfortable with it. By not restricting the subjects in terms of time during training and randomizing the order of scenarios, we aimed to mitigate the learning effect.

After training, human subjects were presented with 27 scenarios, each containing a homogeneous team composed of 3 robots (i.e., all robots possess the same sets of capabilities) and 6 tasks. They were instructed to assign robots to tasks, such that execution time would be minimized, and to consider the following factors:

1. They can spend as much time as needed on a scenario; their performance is computed based only on the efficiency (overall execution time of tasks) of their task assignment.

2. All task types have equal priority.

3. Multi-robot tasks can only be executed when all assigned robots arrive at the task’s location in the environment.

4. Once the required number of robots reaches a task location, then that task is considered complete. In other words, task execution duration is assumed to be negligibly small compared to the travel time incurred by the robots.

5. In order to complete a scenario, all tasks have to be assigned the exact number of robots that they require.

<table>
<thead>
<tr>
<th>category label</th>
<th>APR</th>
<th>CTR</th>
<th>number of scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>low = 1.0</td>
<td>low = 0.0</td>
<td>6</td>
</tr>
<tr>
<td>Q2</td>
<td>low = 1.0</td>
<td>high &gt; 0.0</td>
<td>6</td>
</tr>
<tr>
<td>Q3</td>
<td>high &gt; 1.0</td>
<td>low = 0.0</td>
<td>6</td>
</tr>
<tr>
<td>Q4</td>
<td>high &gt; 1.0</td>
<td>high &gt; 0.0</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 3.1: TCAT experiment scenario categories
Following completion of each scenario, users responded to a quick survey (given on-line, as part of the TCAT interface) to obtain each user’s subjective assessment of the mental demand required by the scenario and their confidence in their own solution and performance. Finally, each experiment ended (after all on-line scenarios and surveys were completed) with an in-person interview, in which subjects were asked to give their opinions on the ease of understanding and use of the interface, features of the scenarios that required higher mental demand, and their general strategies for approaching the task assignment problems given.

Note that the only information given to users to estimate the travel time for each robot was the relative lengths of paths (computed by the A* algorithm [40] as the path planner) that they would follow to reach their goals. This may have introduced a bias in assessing mental demand measurements, especially when the paths were long and contained a high number of turns, which potentially can make visual comparison of alternative paths harder. This was also mentioned as a hindrance by some of the participants during post-experiment interview sessions.

3.5.3 Metrics and Variables

For assessing human subjects’ perceived complexity and confidence in their task assignments, the NASA-TLX [1] workload metrics were adapted. Since the subjects were not required to perform any physical activity and they were not limited in terms of time, the physical and temporal demand components of the standard NASA-TLX were deemed irrelevant. Similarly, the standard measures of effort and frustration were expected to be very low, due to the relative ease and short amount of time required for completing each scenario. Therefore, the full NASA-TLX procedure was not followed. Instead, only two of the workload metrics—mental demand and performance—were used as stand-alone subjective ratings. Mental demand was also compared with an objective metric: the amount of time it took each subject to complete each scenario (i.e., assign all the tasks); this is referred to here as “task assignment completion time”. Similarly perceived performance was
compared with an objective metric: performance score based on the expected completion time of the assignments. Expected completion time represents the *makespan* as referred to in the scheduling literature [72], and it is the objective of the experiments as instructed to the participants.

In summary, four dependent variables were used (mental demand, perceived performance, plan construction time and objective performance score) and two independent variables (*APR* and *CTR*).

The dependent variables, subjective workload and performance metrics, were further modified and adjusted prior to the analysis. To compare these ratings across human subjects, each subject’s ratings were adjusted by shifting each score so that the median rating for that subject was equal to 50. The exact formula for this process for mental demand is as follows:

\[
MD_{ij} = rMD_{ij} - (\text{median}(rMD_i) - 50)
\]

where \(rMD_{ij}\) represents the subjective mental demand rating of participant \(i\) for scenario \(j\), and the \(MD_{ij}\) represents the adjusted rating. Similarly for performance:

\[
P_{ij} = rP_{ij} - (\text{median}(rP_i) - 50)
\]

where \(rP_{ij}\) represents the subjective performance rating of participant \(i\) for scenario \(j\), and the \(P_{ij}\) represents the adjusted rating. By applying this shift, the data was normalized across users, preserving the ranking order and the differences between the rankings provided by each participant. The rationale for the adjustment was based on statements of several participants during post-experiment interview sessions. According to these participants, their initial ratings were not accurate since they haven’t seen enough scenarios at that point, and as a result the remaining ratings were largely determined by these initial, inaccurate ratings. For example, if the first rating that they provide for the mental demand was on the high side for a scenario, which later is found
to be one of the easier ones, they got stuck having to rate the remaining scenarios on a limited portion of the scale.

Since there were no time restrictions for task assignment completion, the raw data for average time across all scenarios varied significantly between participants. In order to be able to compare results based on mental demand variable, for each participant, the task assignment completion times were also adjusted and represented as the percentage of the maximum time they spent on a scenario, labeled $TACT_{pct}$. The computation of $TACT_{pct}$ is as follows:

$$TACT_{pct_{ij}} = \frac{TACT_{ij}}{\max(TACT_i)} \times 100$$

where $TACT_{ij}$ represents the task assignment completion time in seconds, for participant $i$ in scenario $j$ and $TACT_{pct_{ij}}$ represents the percentage of the $TACT_{ij}$ compared to the maximum time spent on a scenario by user $i$.

In order to check if the subjective performance rating was consistent for each user, a score was computed based expected completion times of the assignments. This is used as the objective metric for performance, as described in the beginning of this section. This performance score is computed as follows:

$$P_{score} = \frac{((ECT_{auto} - ECT_{user})/ECT_{auto}) \times 100}$$

where $ECT_{user}$ represents the expected completion time of the user-generated plan and $ECT_{auto}$ represents the expected completion time of an automated solution. This automated solution was

<table>
<thead>
<tr>
<th>Metric</th>
<th>min</th>
<th>max</th>
<th>$\mu$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental demand (raw)</td>
<td>0.77</td>
<td>99.42</td>
<td>45.8</td>
<td>24.24</td>
</tr>
<tr>
<td>Performance (raw)</td>
<td>19.73</td>
<td>100</td>
<td>77.38</td>
<td>16.76</td>
</tr>
<tr>
<td>$TACT$ (seconds)</td>
<td>20.4</td>
<td>1230.8</td>
<td>103.46</td>
<td>104.79</td>
</tr>
<tr>
<td>Mental demand (adjusted)</td>
<td>-12.26</td>
<td>96.93</td>
<td>48.06</td>
<td>18.72</td>
</tr>
<tr>
<td>Performance (adjusted)</td>
<td>-4.21</td>
<td>89.27</td>
<td>48.15</td>
<td>14.22</td>
</tr>
<tr>
<td>$TACT_{pct}$</td>
<td>6.75</td>
<td>100</td>
<td>42.0</td>
<td>21.45</td>
</tr>
</tbody>
</table>

Table 3.2: Mental demand, Performance and Task Assignment Completion Time statistics, before and after adjustments

A total of 14 human subjects participated in the experiment. The participant population consisted of undergraduate and graduate students as well as faculty, all from technical fields. The participants’ age and experience levels ranged as shown in Table 3.3. Participation was strictly voluntary, and subjects were not paid.

For statistical analysis, a Two-Way Repeated Measures Analysis of Variance (ANOVA) test was used with the independent variables \( APR \) (low, high) and \( CTR \) (low, high). Since for each participant and condition, the data contained multiple observations (e.g., 6 observations for participant 1 in Q1. see Table 3.1), during the analysis the mean of these observations are used. All four dependent variables met the conditions of normality and homogeneity. The values for mental demand and perceived performance were entered by the human subjects in on-line post-scenario surveys. The summary statistics are listed in Table 3.4.

The effects of both independent variables were found to be statistically significant on mental demand: for \( APR \), \( F(1,13) = 110.8 \) and \( p < 0.001 \); for \( CTR \), \( F(1,13) = 115.8 \) and \( p < 0.001 \). Also, the interaction of \( APR \) and \( CTR \) was found to have a significant effect on mental demand,
$APR \times CTR$, $F(1,13) = 10.21$ and $p = 0.007$. The box-plot for the mental demand distributions for each scenario category can be seen in Figure 3.5. There is a strong correlation ($r(12) = 0.51$, $p < 0.001$) between mental demand and $APR$, therefore we can reject the null hypothesis and accept our alternate hypothesis: increase in $APR$ results in higher mental demand for the experimental scenarios tested. This can be seen in the figure, where quadrants Q3 and Q4 have higher means than Q1 and Q2, respectively, each containing only single-robot tasks.

As for the second hypothesis, there is a moderate negative correlation ($r(12) = -0.213$, $p < 0.001$) between mental demand and critical task ratio. This was surprising since our expectation was that critical tasks would require more mental effort. A possible explanation may be the reduced search space, as most of our subjects indicated during their post interviews: once tasks were determined to be critical, they were given higher priority, which made it possible to consider fewer alternative solutions for the remaining tasks; thus resulting in overall reduced mental workload. Although not strong, the statistically significant correlation leads to rejection of the null hypothesis and because it behaves exactly opposite of what was expected, it invalidates our hypothesis as well. This can be seen by comparing Q2 to Q1 and Q4 to Q3. We speculate that this behavior pattern may not hold if missions are critical or costly, or if scenarios are so difficult that the solution space is less easily reduced.

The effects of both independent variables were found to be statistically significant also on perceived performance: for $APR$, $F(1,13) = 38.76$ and $p < 0.001$; and for $CTR$, $F(1,13) = 43.5$ and $p < 0.001$. However the interaction of $APR$ and $CTR$ was found not to have a significant effect on perceived performance, $APR \times CTR$, $F(1,13) = 0.04$ and $p = 0.844$. Results are shown in Figure 3.6. Perceived performance produces opposite behavior to mental demand, which can be observed by comparing Q3 to Q1 and Q4 to Q2 for $APR$, and Q2 to Q1 and Q4 to Q3 for $CTR$. These results can also be seen from the marginally strong correlation between mental demand and perceived performance ($r(12) = -0.45$, $p < 0.001$), suggesting that higher mental demand yields
To check if the mental demand rating was consistent for each user, the task assignment completion time was chosen as an objective metric for mental workload. The effects of both independent variables were found to be statistically significant on $TACT_{pet}$: for $APR$, $F(1, 13) = 83.02$, $p < 0.001$, and for $CTR$, $F(1, 13) = 9.56$, $p = 0.008$. As for the interaction of $APR$ and
CTR, similar to the perceived performance case, it was found not to have a significant effect on $TACT_{\text{pct}}$, $APR \times CTR$, $F(1, 13) = 3.211$ and $p = 0.065$. The results are shown in Figure 3.7. The pattern is similar to that of mental demand, where a strong correlation ($r(12) = 0.5$, $p < 0.001$) with $TACT_{\text{pct}}$ exists.

![Figure 3.7: Task assignment completion times (represented as % of maximum task assignment completion time across runs, adjusted for each subject) by Scenario categories](image)

The main effect of $APR$ is found to be statistically significant on objective performance: $F(1, 13) = 160.6$, $p < 0.001$. However no significant effect was found for $CTR$, $F(1, 13) = 2.02$, $p = 0.18$ and the interaction of $APR$ and $CTR$, $APR \times CTR$: $F(1, 13) = 0.02$ and $p = 0.889$.

These analysis on the objective performance($P_{\text{score}}$), reveal an interesting result, when compared to the perceived performance. First, for the scenarios that are presented, there is almost no difference between the categories for $Q1$ and $Q2$ and also for $Q3$ and $Q4$, as can be seen in Figure 3.8, which suggests that $CTR$ does not affect objective performance as also found in the ANOVA analysis. This result contradicts the findings in the perceived performance results where $CTR$ has a significant effect. Further correlation analysis support this interesting result, that there is no significant correlation found between the subjective performance ratings and the $P_{\text{score}}$ ($r(12) = -0.08$,
Based on these analysis, it can be said that the users feel better about their performance when CTR is high, probably due to reduced search space, when there is no significant difference in their actual performance. The main difference is seen as a result of APR where the objective performance drops significantly due to increased variance.

Figure 3.8: Objective performance scores by Scenario categories

The results for mental demand show that in the dimension SR – MR, it becomes harder for operators to devise a plan as APR increases. One observation highlighted during the post-scenario interviews suggests that when multi-robot tasks require all the robots, as opposed to only some of the robots, it is easier to devise a plan. This also supports our previous pilot study findings regarding the reduced mental demand when task assignment options are relatively limited. In the IT – CT dimension, the results show that in the presence of critical tasks, it becomes easier for operators to devise plans.

Since all independent variables only have two factors (high, low), post-hoc analysis was not necessary.
Analysis with One-Way Repeated Measures ANOVA

As an alternative, it is possible to perform the analysis of the TCAT experiment data with one independent variable representing the quadrants defined in Table 3.1, therefore with values (Q1, Q2, Q3, Q4). Since the underlying factors for the quadrants are APR and CTR, we expect to see similar results with our Two-Way Repeated Measures ANOVA analysis. Therefore, in order to check our findings in our previous analysis, we have performed a One-Way Repeated Measures ANOVA, with Quadrant as our independent variable and all four dependent variables, mental demand, perceived performance, TACT pcf and P score.

The results of the analysis showed that the main effect of Quadrant was found to have a significant effect on all dependent variables.

For subjective workload, mental demand, $F(3, 39) = 43.98, p < 0.001$,
For perceived performance, $F(3, 39) = 19.95, p < 0.001$,
The objective workload, TACT pcf, $F(3, 39) = 40.93, p < 0.001$, and
The objective performance, P score, $F(3, 39) = 43.87, p < 0.001$.

In order to find out which of the levels of the Quadrant have an effect on the dependent variables, a follow up post-hoc analysis was conducted using Pairwise T-test with Bonferonni Correction. The results of these analysis is presented in Table 3.5.

According to these results, for mental demand all pairwise level means are found to be significantly different, similar to our Two-Way Repeated Measures ANOVA results (see Figure 3.5).

Similarly, for objective workload, TACT pcf, all level means were found to be significantly different with the exception of the difference between the means of levels, Q3 and Q4. This result states that CTR has no significant effect on objective performance, when APR category is high. Although we can’t make the same statement when the APR is low, based on the non-significant interaction between APR $\times$ CTR in the Two-Way Repeated Measures ANOVA analysis, we can conclude that the difference between groups Q1 and Q2 is not high enough to make the interaction
Mental demand | $T_{ACT\_pct}$
---|---
Q1 | Q2 | Q3 | Q1 | Q2 | Q3 | Q1 | Q2 | Q3
Q2 | $< 0.001$ | − | − | Q2 | $= 0.007$ | − | −
Q3 | $< 0.001$ | $< 0.001$ | − | Q3 | $< 0.001$ | $< 0.001$ | −
Q4 | $< 0.001$ | $< 0.001$ | $= 0.0092$ | Q4 | $< 0.001$ | $< 0.001$ | $= 1.0$

| Perceived Performance | $P_{score}$
---|---
Q1 | Q2 | Q3 | Q1 | Q2 | Q3 | Q1 | Q2 | Q3
Q2 | $= 0.0013$ | − | − | Q2 | $= 1.0$ | − | −
Q3 | $< 0.001$ | $< 0.001$ | − | Q3 | $< 0.001$ | $< 0.001$ | −
Q4 | $= 1.0$ | $< 0.001$ | $= 0.0011$ | Q4 | $< 0.001$ | $< 0.001$ | $= 1.0$

Table 3.5: Pairwise T-tests.

The group means are significantly different for perceived performance, except for levels Q1 and Q4 (see Figure 3.6). According to this result there is no significant difference between subjective performance ratings of groups with $(APR, CTR)$ values $(\text{low, low})$ and $(\text{high, high})$. The same result, is observed in the non-significant interaction between $APR \times CTR$ in the Two-Way Repeated Measures ANOVA analysis.

Finally, the group means are significantly different between Q1 and Q2 and also between Q3 and Q4 (see Figure 3.8). This result suggests that $CTR$ has no effect on the objective performance $P_{score}$, which is also confirmed by our Two-Way Repeated Measures ANOVA analysis.

### 3.5.5 Post-experiment Interviews

After completion of all the scenarios, the participants were asked three questions during a post-experiment interview session:

1. How was your experience with the tool? Was it easy or hard to work with? Any comments about your experience?

2. After going through all the different scenario types, which one’s do you think stand out as harder ones?

3. Did you had a particular strategy for allocating the tasks? When you encountered a scenario
for the first time did you started allocation the same way, or did you have several different strategies for different situations?

Participants were asked the first question in order to determine if the design of the interface had a negative impact. All participants stated that the interface was intuitive and easy to use, and that they did not face any significant difficulties in constructing plans.

For the second question, when the participants were asked to name the features or define conditions that made certain scenarios harder to solve than others, the answers varied. The majority of participants stated that mental effort was significantly greater when faced with multi-robot task scenarios. Some stated that was because they were motivated to minimize “idle” time for robots, i.e., arrange the assignments so that robots would rendezvous at task locations at approximately the same time. A few participants mentioned that decision making was harder in scenarios where robots were equally capable of reaching tasks that were distributed around the environment.

For the last question, participants were asked to describe their approaches and strategies for problem solving. The answers to this question can be classified in three groups:

- participants first assigned single robots to tasks that were close to robots’ locations and worked their way to a complete solution using a greedy approach;

- critical multi-robot tasks were given high priority and considered first, then the remaining tasks were assigned after a plan was formed for the high priority tasks; and

- the environment was divided into geographic sectors and robots were assigned to tasks within their sectors.

I’ll revisit these answers in Chapter 5, when discussing integration of human strategies into decision support tools.
3.6 Summary

In this chapter, I described the task landscape that the MAS and MRS communities focus on. As opposed to the existing taxonomies which are devised to define the computational complexity of a task assignment problem, I discussed the same problem from the human operator’s perspective. To this end, four task properties; requirement, constraints, frequency and granularity were identified and discussed in terms of their effects on the mental workload of the operators. Among these, requirement and constraints are deemed to be the most important characteristics that commonly appear in MRTA problems and a graph-based model, TAG is devised to capture these properties.

To verify the metrics derived from the TAG model, I have developed the TCAT, and a user study was conducted presenting Urban Search and Rescue (USAR)-inspired scenarios to participants. In these studies, two metrics, \( APR \) and \( CTR \), representing requirement and constraint properties of the tasks, respectively, are studied. Results show that \( APR \) is a significant factor affecting mental workload and perceived performance, as expected. Results also show that \( CTR \) surprisingly helps solving problems due to the reduced number of choices available for assignment, however this is only marginally statistically significant (\( 0.05 > p > 0.001 \)), contrary to the statements made by most subjects.

The TCAT tool was first demonstrated in [69], the TAG model first appeared in [70] and the analysis of the results of the experiment is presented in [71].

These results answer research question RQ3 that is posed in Chapter 1.2, for the two factors, requirement and constraint properties of tasks. In order to feasibly analyze the effects of the chosen parameters, the effects of frequency and granularity is left for future work. As RQ3 states, the requirement and constraints features are verified in favorable circumstances, where participants were not subjected to any time pressure.

In the next chapter, I look into how distribution of the platforms (resources) and tasks in the spatial map affect performance and mental workload of the operators in dynamic real-time
environments.
Chapter 4

MRTA Complexity Analysis in Real-Time and Dynamic Environments

In the previous chapter, we investigated human approaches to task assignment problems in a static environment where the users were not expected to follow through with the execution of their plans. This limitation helped us to feasibly conduct a human subject experiment with a relatively large number of scenarios, in order to verify the TAG model under varying circumstances such as different map topologies. In this experiment, the users were instructed to come up with the most effective plan, in terms of expected execution time and to take as much time as necessary to construct these plans, allowing us to observe human approaches to task assignment problems in a stress-free environment.

In application domains that are of interest to this work, these limitations and approaches are not realistic and the problem requires a holistic approach, where the operators’ role is not only limited to task assignment but also observing the execution of the plan and modifying it when necessary, in response to the changes in the environment and the state of the individual robots.
on the team. These extended responsibilities place additional cognitive workload on the operator, impacting the operators’ situational awareness, which according to my thesis statement, can be remedied to some extent with the use of automated task assignment planners.

In this work, I am investigating usage of automated approaches as a decision support functionality but even in this relatively passive role, in order for these systems to be of measurable help to the operator, the suggested plans must be similar to the expectations of the human users, otherwise, it will be hard for users to understand the suggested plans, and gaining that extra understanding will increase users’ cognitive workload.

As stated in Chapter 1, earlier work suggests [49, 10] that when users don’t trust or fully understand the inner workings of automated systems, they discard any solution computed by them very quickly even if the offered solution is effective. Finding the right balance between the search for the best solution and the ones that would be approved by the human operator is partly a problem of establishing a correct and accurate degree of trust in the autonomous capabilities of the planner. My point of view in this dissertation is that this can best be achieved by a good grasp of how automated planners work and providing human operators with a toolbox of behaviors that they can use to steer the planner to a desired solution.

To this end, to be able to design systems that can be useful, general human expectations and approaches should be understood and modeled in a formal and generalizable manner. Such models are necessary in order to be able to predict situations that will lead to degradation in human performance and situational awareness, which then can be used for comparison of interaction schemes. In the previous chapter, our graph-based model, the TAG was described, which focuses on spatial relationships of resources and tasks. In that chapter, I presented the results from our user study in which two metrics computed from the TAG ($APR$ and $CTR$) were verified as good indicators of human subjects’ levels of mental demand and perceived performance. $APR$ and $CTR$ together encapsulate elements of two complexity dimensions: SR-MR and ID-CN (as discussed
in Chapter 3). Also, TCAT experiment not surprisingly showed that one of the most common approaches to task assignment is a greedy approach, by allocating closest tasks first. At the time of conducting the TCAT experiment, the TAG lacked the travel distance/time information from robots to tasks and between tasks.

In this chapter, I focus on the effects of factors common in dynamic domains. First a formal representation of the assignment problem for the entire duration of the mission is introduced, building on the TAG model. Next, the TASC interface is described, which is similar to the TCAT interface, redesigned for dynamic control scenarios and also containing an automated assignment tool. Finally, results of a user study with the TASC interface is presented, which verifies the extended mission model.

4.1 Extending TAG for Real-Time and Dynamic Environments

A TAG models an isolated assignment problem instance in the SR-MR and ID-CN dimensions. During a mission, the environment or the state of the robots may change, which may in turn force revision of the existing plan. This distinction is represented in [54] by an additional dimension; static vs. dynamic allocation environment (SA, DA). This chapter focuses on situations and environments which fall into the DA category, whereas the previous chapter focused on the SA category.

In a DA domain, planning decisions can be affected by more than just the information that is available at the time of assignment. Without explicit information about future tasks, these factors include utilization of resources, which can be one of two types.

The first type of resource is the robot platforms. The way they are utilized for the tasks at hand dictates their availability and state for future tasks. Since the positions of the robots, after the immediate plan is completed, affects choices that can be made for future task assignments, a human operator may choose to make a sub-optimal plan that would position robots better for engagement in future tasks. Similarly, for the same reason, a operator may choose to use a subset
of the robot platforms that are available and hold the rest of them in reserve, in order to respond better to future tasks.

The second type of resource is the amount of time available for deliberation on the part of the operator, which is a function of factors such as the rate of environmental change which requires attention, number of tasks to be distributed among the robots, effectiveness of the interface and the amount of time that the operator is willing to spend on deliberation about future situations. As described in Chapter 2.1, situations where a human operator can reason about the future requires a high level situational awareness (level 3). Even if the operator has a high level of situational awareness, s/he may choose not to act based on uncertain future projection of events.

Both of these factors vary depending on the skills, experience and other personal characteristics of the operator, which are examined in the TASC experiment described in this chapter. But first, I’ll present an extension to the TAG definition, in order to encapsulate the travel time information between task vertices as well as the travel time between robots and tasks. Then, I’ll formally describe the planning problem for the duration of the mission and metrics to capture the effects of changes in the environment on the decision-making capabilities of the operators.

In order to incorporate the travel distances between tasks, it is sufficient to extend the TAG as a weighted graph, where each edge is associated with a weight, \( w_{ij} \), representing the distance between the center locations of tasks \( t_i \) and \( t_j \) respectively. Incorporating the travel cost of robots to tasks requires an additional vector, \( Cost_i \), for each task vertex, \( v_i \) where \( Cost_{ij} \) represents the cost of robot \( r_j \in Acc_i \) reaching the task \( t_i \). In the MAS and MRS literature, the term travel cost can incorporate many factors, such as energy consumption of the robot, road condition terrain, and most importantly, the distance between the robot and the task. In this work, since the focus is on scheduling, I’ll refer to \( Cost_{ij} \) as the travel time of robot \( r_i \) to reach task \( t_j \).

We’ll refer to the assignment problem as the \( MAP^1 \) which is essentially an ordered list of

\[ \text{MAP}^1 \]

\(^1\)From this point on in the dissertation, I’ll use the MAP abbreviation to refer to this model, and use lowercase “map” exclusively to refer to the geographic map.
TAGs:

\[ MAP = \langle TAG_{t_0}, TAG_{t_1}, ..., TAG_{t_k} \rangle \]

where \( t_i \) represents the time the TAG is added to the MAP, \( t_0 \) represents the time when the first task, \( t_k \) represents the time when the last task appears during the mission and \( t_i < t_{i+1} \) holds for all \( i \).

Every entry in the MAP represents a decision point regarding the assignment that is required due to a change in the spatial relationships between tasks and accessibilities of the robots. A change like this occurs when:

i. a new task appears, an existing task is removed or gets completed,

ii. a new robot is introduced into the environment or an existing robot becomes unusable,

iii. properties of an existing task changes, and/or

iv. additional obstacles appear or known obstacles disappear in the environment, changing the accessibilities of robots and tasks.

While items (i) and (iv) cause a change in both the graph topology of a TAG and the \( \text{Dom}, \text{Acc} \) and \( \text{Cost} \) values of its nodes, items (ii) and (iii) change only the \( \text{Dom}, \text{Acc} \) and \( \text{Cost} \) values. Although completion of a task may mean a significant change between any two consecutive TAGs, from the standpoint of assignment planning, it is not an unexpected change, therefore not considered to be a decision point, where the user has to update the existing plan.

An example of how the MAP changes during a mission is depicted in Figures 4.1, 4.2 and 4.3. In the first, Figure 4.1, two of the robots are working on a critical task that blocks the corridor. Upon completion of the task the situation becomes as shown in Figure 4.2.

Also it is important to note that when robots are in motion, the \( \text{Cost} \) values of the task vertices will be in constant change and it is safe to say that this change may trigger replanning.
by the human operator. A situation like this may arise if the robots’ actual trajectories differ from their initially planned trajectories, for reasons such as collision avoidance, which may position robots in locations where it would be more convenient for them to execute tasks that are not their immediate target. However, it is hard to predict when such replanning may be triggered since it is highly dependent on the human operator and the domain.

Figure 4.1: A TAG, where robots $r_1$ and $r_2$ are working on task $t_4$.

4.1.1 Metrics

As the TCAT experiment verified, high APR values increase the mental workload of the operator and high CTR values, inversely, reduce the workload. While the APR metric represents the portion of the tasks that require close coordination, in TASC, I look for verification of both of these values in dynamic settings. In addition, I investigate a third factor, the spatial distribution of the robot platforms in the environment. This factor was not studied in TCAT, mainly because it would otherwise increase the number of scenarios to the extent that it would be unfeasible to conduct the
study in a reasonable amount of time for each participant, given the number of scenarios that we were testing. The purpose of the TASC user study is to test this model in DA environments.

One of the goals for defining the MAP model is to devise valid predictors for operator performance and mental workload. Essentially, my hypothesis is that both of these outcome variables are directly affected by the number of solutions that can be produced for the problem, which in turn is affected by:

i. The ratio of tasks that require close coordination to the total number of tasks (MR-SR dimension),

ii. the critical nature of the tasks (ID-CN dimension), and/or

iii. the spatial distribution of the robot platforms across the map.

In addition to the factors affecting the number of possible solutions,

iv. due to the dynamic nature of the domain (SA-DA dimension), changes in the environment
may force operators to revise their completed assignments, adding an extra mental workload on the operator which may potentially lead to worse performance.

In this chapter, I am introducing two new metrics to consider:

- **Average Domain Density (ADD)**, which is \( \text{ADD} = \frac{\sum_{i=1}^{n} DD_i}{n} \), where,

\[
DD_i = \frac{|\text{Dom}_i|}{\binom{m}{\text{req}_i}}
\]

In the above equation, \( DD_i \) represents the domain density of vertex, \( v_i \). Domain density for a vertex, \( v_i \), is calculated by the dividing the size its current domain by the potential size of the domain if all robots were able to access to \( v_i \).

- **TAG Disruption Ratio (TDR)**, which is \( \text{TDR} = \frac{|A_i| - |A_{i+1}|}{|A_i|} \), where, \( A_i \) is the set of assignments performed in \( TAG_i \), just before \( TAG_{i+1} \) and \( A_{i+1} \) is the number of assignments that are still valid, right after the arrival of \( TAG_{i+1} \). In other words, TDR is the ratio of the
number of assignments removed to number of assignments performed before the arrival of the new TAG.

The ADD accounts for the condition (iii.), while TDR accounts for the condition (iv.).

4.1.2 MAP adaptation into TASC experiment

For the purposes of the TASC experiment, I am only interested in situations that would guarantee replanning, therefore I populate the MAP with TAGs that are only generated when a new task appears. Also, in the TASC experiment, the number of robots is constant, therefore the MAP will not contain any TAGs generated due to changes in the number of robots on the team. As can be seen from Figures 4.1 and 4.2, the TAGs of these two situations are significantly different from one another, as a result of the completion of task $t_4$. However, the TAG in Figure 4.2 will not be added to the MAP for the scenarios of the TASC experiment. The rationale for this is that the assignment to execute the task was performed by the operator or with the approval of the operator, therefore these changes are expected and so not likely to initiate a reconsideration of the existing plan from the perspective of the operator. Therefore, any TAGs resulting from completion of a task will not be included in the TASC experiment. Similarly, as mentioned in the previous section, the Cost vector constantly changes as robots move around in the environment and it is hard to predict when human operators will react and replan as a result of these changes. Since, there is no guarantee that a replanning event will occur, no TAG is generated and added to the MAP in the TASC experiment.

Before I discuss the user study that is conducted to verify the extended TAG and MAP models, I will describe the testbed in the next section.
4.2 Task Assignment Supervisory Control (TASC) Interface and Testbed

Although the main focus of this work is not interface or interaction design, in order to be able to interpret the results of the experiment accurately, it is necessary to briefly explain the TASC testbed and the HRTeam framework [89, 28] that it interacts with, in order to communicate with simulated robot platforms.

The TASC interface is built as an extension to the TCAT interface, described in Chapter 3.4. The interaction to add and remove assignments of robots to tasks is exactly the same in both interfaces. The TASC interface has additional visual components to provide information related to the dynamic nature of the environment. As can be seen from Figure 4.4, the TASC interface has the same Map and the Allocation components as TCAT. The main difference is the introduction of the Timeline component (displayed underneath the Map component), which shows the expected execution of the assignment and a clock showing the time passed since the start of the experiment. Also there are three buttons in the TASC interface: the wait button to halt the robots; the clear button to remove all assignments; and the auto assign button which is available only in certain modes of operation. The full functionality of the auto assign feature will be discussed in Chapter 4.2.3.

The TASC interface communicates with the physical or simulated robots through a set of messages. The robots’ locations, paths to their targets and their state information are updated in the interface, based on the messages sent by the robots. All task assignments and removals are made via the TASC interface and are immediately declared to the robots. In this setup, the robots have complete autonomy in terms of path planning and navigation as well as collision avoidance, while, they rely fully on the TASC interface to dictate which tasks that they will execute and the order in which they will execute them.

There are 3 modes of operation of the TASC interface, which I will discuss in detail in the
4.2.1 Manual Allocation Mode

In the manual allocation mode, as the name implies, the users perform all assignments manually. Assignments are performed on the allocation component (the component on the right in Figure 4.4) of the TASC, by first selecting the robot icon(s) and then selecting a task icon. Assignments are then immediately conveyed to the robots via an `ADD_ASSIGNMENT` message. Upon receiving such a message, task information is added to the end of a robot’s list of tasks that it needs to execute. This list is referred to as a robot’s `agenda`. Robots do not wait for an additional command to move towards the first task in their agenda; they start to move there as soon as the task is added.

Any removal action is also conveyed immediately via a `REMOVE_ASSIGNMENT` message to the robots, which they respond to by removing the task item from their agenda. If the removed task was the first in their agenda, they change target and move towards the next task item in their agenda, if such a task exists. The removal of an assignment can be performed on the allocation component in one of three ways: (i) by selecting an assignment, represented as a line between
a robot and a task, (ii.) by selecting a robot, (iii.) by selecting a task and hitting the DELETE key on the keyboard. The first way (i.) only removes a single assignment, while items (ii.) and (iii.) remove all of the selected robot’s assigned tasks and remove all of the robot assignments to the selected task, respectively. Alternatively, the CLEAR button located under the allocation component, removes all of the assignments. The REMOVE_ASSIGNMENT message is disregarded only if the robot has already started executing the task that is commanded to be removed.

In terms of Sheridan’s LOA, this interaction in manual allocation mode corresponds to level 1, where the computer offers no assistance and to supervisory interaction according to Hearst’s interaction scale, described in Chapter 2.1.

4.2.2 Automated Allocation Mode

The automated allocation mode refers to full autonomy in terms of task assignment and corresponds to level 10 in Sheridan’s LOA, where there is no interaction with a human operator. This mode is used for comparing different MRTA allocation methods as well as serving as a benchmark for evaluating the performance of human subjects. Comparisons of various methodologies and heuristics will be examined later in Chapter 5.

For evaluation of human subject performance, the adopted approach for the automation is inspired by the SSI auction as described in Chapter 2.2.1. In SSI, all items are auctioned at the same time and bidders send a bid for every item. Once the auctioneer receives all bids, it searches for the highest bid among all auctions. Upon finding the highest bid, it clears only the auction that the highest bid is submitted to and allocates the item to the bid owner. The remaining items are then auctioned again and the bidders send new bids to these items and the process repeats until all items are allocated. The main advantage of clearing one auction at a time is that in the later rounds of auctions, the winners of the earlier rounds have full knowledge of which items they have won therefore can update their valuations regarding the remaining items. The main disadvantage
of SSI is the high communication complexity, $O(nm^2)$, where $n$ is the number of robots and $m$ is the number of tasks. SSI is considered as an alternative to combinatorial auctions. While combinatorial auctions allow bidding on bundles of items, and therefore are expected to provide optimal allocations, they suffer from the computational complexity of winner determination and item bundling problem which are both NP-hard. SSI auctions are used in MRTA problems as an alternative to combinatorial auctions, due to their simplicity, scalability and acceptable solution quality [51].

As covered in Chapter 2.2.1, auctions are used in MRTA problems such as exploration to distribute unvisited map coordinates to robots, with the intention to minimize the overall cost based on distances and other costs incurred by the robots for travelling to these coordinates. Similarly, in this work I use the tasks’ locations as auction items and bids as distances of the robots to these tasks.

In automated allocation mode, I use a centralized task allocation algorithm (see Figure 4.5), which behaves like the clearing mechanism of an SSI auction. According to this algorithm first two sets ($V_{\text{open}}$ and $V_{\text{allocated}}$) are generated, holding the copies of unallocated and allocated task vertices in the TAG. Initially, $V_{\text{open}}$ only contains task vertices that have non-empty domains. Therefore, the task with the minimum earliest start time ($est$) value is allocated at every iteration. This happens in line 4. The $est$ value for task vertex $v_i$ is computed as follows:

$$est_i = \min \{DCost_{ij}\}_{j=1}^{\text{Dom}_i}$$

$$DCost_{ij} = \max \{Cost_{ik}\mid r_k \in \text{Dom}_{ij}\}$$

where $Cost_{ik}$ represents the cost of robot $r_k$ reaching task vertex $v_i$, and $DCost_{ij}$ represents the maximum arrival cost among all robots in a domain element $j$. This equation also states that the tasks can only begin upon arrival of all assigned tasks.
1: $V_{open} \leftarrow \{v_i|Dom_i \neq \emptyset\}$
2: $V_{assigned} \leftarrow \emptyset$
3: while $V_{open} \neq \emptyset$ do
4: $v_a \leftarrow \min\{v_i|est_i \leq est_j, \forall v_i, v_j \in V_{open}\}$
5: for all $r_k \in Dom_a$ do
6: $agenda_{r_k} \leftarrow v_a$
7: end for
8: $V_{assigned} \leftarrow v_a$
9: $V_{open} \leftarrow V_{open} - v_a$
10: recomputeCosts($V_{open}, V_{assigned}$)
11: end while

Figure 4.5: Auto assign algorithm

Once the allocation of robots to the task $t_i$, represented by the processed task vertex $v_i$, is done, the TAG is updated by the recomputeCost procedure on line 10. This procedure essentially updates the $Acc$, $Dom$ and $Cost$ vectors of every task vertex except $v_i$, as if the allocated task, $t_i$, is executed and completed. This process is repeated until all tasks are assigned to robots.

The reason for using a centralized task allocation algorithm was mostly for practical implementation reasons and since the number of tasks and robots were small it wasn’t deemed critical to use the SSI auction itself. But it is obvious to see that this algorithm can be replicated as an SSI auction by the updating the $Cost$ vector of any given TAG based on robots’ bids rather than computing the distances between robots and tasks centrally.

This algorithm can be triggered at any point during mission time, whenever replanning is deemed necessary. In the TASC experiment, this replanning is dependent on the generation of a new TAG, which only happens upon arrival of new tasks.

4.2.3 Collaborative Allocation Mode

In collaborative allocation mode, the TASC interface provides an optional button, AUTO ASSIGN, for the user to delegate the task assignment process to the automated solver, which is the same method described in the previous section (4.2.2). This option becomes enabled when there are unassigned tasks and becomes disabled when all tasks are assigned. Furthermore, this option does
not correct or modify existing assignments. As soon as this option is triggered the system will allocate unassigned tasks without waiting for the users approval. The users however, can modify the assignments made by the AUTO ASSIGN tool afterwards.

This interaction can be interpreted as roughly corresponding to level 5 or level 7 of Sheridan’s LOA scale. Since the tool does not require an explicit approval from the user once the assignment of tasks is delegated, it might be considered as level 7. However, it is important to note that this tool does not provide a solution unless the human operator commands the system to do so, which is different than system taking the initiative and making the assignment as level 7 LOA suggests.

As stated, the main goal for developing the TASC tool is to conduct human-subject experiments, therefore it was designed to present a series of scenarios interleaved with surveys for collecting subjective assessment from users regarding the situations faced during the most recently completed scenario. These surveys include a complete version of NASA-TLX as well as other custom ones. These surveys and other screens of the TASC interface can be seen in Appendix B.

4.3 HRTeam Framework

HRTeam framework is an experimental testbed for conducting MRS and HRI experiments both with simulation and physical robots [89, 28]. The general architecture of the system can be seen in Figure 4.6. This framework, at the low level uses the Player/Stage [33] system to interface with the physical robots as well as the Stage simulator. The components of the system communicate with each other using the CentralServer process, which acts as a message router for the overall system. Each individual robot is controlled by a RobotController process which governs the autonomous and low-level actions of the robots. In this framework, the TASC interface acts as the OperatorInterface. The other components, CameraAgent, FORREngine and physical robots are not used in the TASC experiment.
The autonomous behaviors of the robots, as mentioned earlier, are limited to path planning, navigation, collision detection and avoidance. For path planning, a standard A* algorithm [40] is used on a navigation graph, which is generated based on the provided map or floor plan. Aside from the path planning, A* is also used for computation of extended TAG weights and travel times of robots in Cost vectors of each vertex.

Collision detection and avoidance are handled via a message protocol. The CentralServer is designed to request periodic position updates from the robots, which is then forwarded to the components that make use of this location information. My collision detection and avoidance protocol works based on these periodic position updates received from robots and broadcast to the entire team by the CentralServer. According to this protocol, each robot keeps track of its teammates’ locations and initiates a collision avoidance sequence with any robot that is in close proximity to itself (defined by a fixed threshold). Any teammate that is located within a fixed distance threshold from a robot is considered to be in that robot’s collision zone. Furthermore, any teammate within a robot’s collision zone is considered to be also on a collision course with the robot if the teammate is located where the robot is heading (defined by a fixed angle threshold). Robots that do not have any teammate in their collision course ignore any communication regarding a
collision and continue moving along their paths. However, if a teammate is on a collision course with a robot, first the robot computes an alternative path to its target considering the area occupied by its teammate as an obstacle. Then, it sends a message to the teammate, which contains the difference between the lengths of the newly computed alternative path and the original path. This message represents the additional cost for the robot to execute the alternative path. If robots receive a cost message from their teammates, they decide to either wait or to assume the right of way, if their own cost is lower than their teammates. This protocol, having no centralized decision maker, is subject to deadlocks which are detected by a cycle detection mechanism and resolved based on robot id.

One important feature of the collision detection and avoidance protocol is that, when initiated, one party almost always ends up waiting for the other robot, that has the right of way, to move to a position out of the collision zone. In the TASC experiment, this feature shows itself as a delay in the execution of planned paths, therefore changing the expected execution time of the plan. The implications of this behavior on the human subjects are discussed later, in Chapter 4.4.4.

4.4 TASC User Study

The TASC interface was used to conduct a user study which had two goals: to verify the factors, represented by metrics derived from extended TAG model, for dynamic environments, which is covered in this chapter; and to observe human operators’ varying responses and approaches to the situations that commonly arise in these domains, which is covered in Chapter 5. In this study, the participants are presented with a set of dynamic scenarios, where the users are expected to plan a task assignment. These scenarios can be seen in Appendix F. In half of these scenarios, the users were given the option to work collaboratively with an autonomous planner as described in Chapter 4.2.3.

The first goal of this user study is to verify the MAP model with the minimum set of
factors that represents the human mental workload, which can be computed in real-time missions. Essentially, if the number of possible solutions to any given instance of a task scheduling problem could be computed in real-time, it would be possible to predict the mental workload, based on the assumption that a higher number of possible solutions require higher mental workload. But since the GAP is NP-complete (see Chapter 2.4), it is necessary to utilize a simple model to predict the mental workload and situational awareness degradation that results in significant performance loss.

Therefore the analysis in this chapter is focused on the verification of the factors. If the factors investigated making up the model actually have an effect on SA or performance and can be determined in runtime in a scalable manner, then they can be used as a model to predict SA degradation and can be used to base interaction design-related decisions in future.

For the first goal, every decision point in a MAP is analyzed based on the TAG metrics representing the three main dimensions:

1. percentage of assignments that require close coordination (MR-SR),

2. the critical nature of the tasks (ID-CN), and

3. the spatial distribution of the robot platforms across the map.

My general hypothesis is that the these factors are sufficient to explain the variance in the mental workload for the task assignment problem within the investigated domain. In order to test this thesis, each plan generated by the human participants is analyzed based on the plan generation time, which represents the mental workload. Plan generation time, in this context, is the same as Percent Busy Time mentioned in Chapter 2.3, which is known to be an effective proxy for mental workload. Also, the effectiveness of the human-generated plans is compared to an autonomous solution in order to determine the task effectiveness, TE, as described in Chapter 2.1.1, which is specified as the makespan or the expected completion time of the plan.

Specifically I expect the following hypotheses to hold in TAG-level:
Table 4.1: TASC Participant Demographics.

<table>
<thead>
<tr>
<th>Age (years)</th>
<th>Gender</th>
<th>Computer experience (years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>µ</td>
<td>σ</td>
<td>Male</td>
</tr>
<tr>
<td>31.5</td>
<td>10.33</td>
<td>20</td>
</tr>
</tbody>
</table>

1. *ADD* has an effect on the cognitive workload and the performance

2. *CTR* has an effect on the cognitive workload and the performance

3. *TDR* has an effect on the cognitive workload and the performance of the operator during task assignment planning.

4.4.1 Experimental Setup

Overall, 30 participants were recruited for the user study and each were paid US$25. The experiment lasted approximately 2 hours per person. The demographic information about the participants is provided in Table 4.1.

The experiments were run on two laptops with the following specifications:

I. 2.00 GHz dual-core Intel T7200 processor with 2GB RAM

II. 2.50 GHz quad-core Intel i53210M processor with 8GB RAM

Laptop I, was used to run Player/Stage simulator, as well as the CentralServer and a RobotController process for each robot in the experiment. Laptop II was used to run the TASC interface. Running the Stage simulator, RobotController and CentralServer processes on a single machine was mainly a setup decision, in order to prevent system performance related issues on laptop II.

The experiment scenarios were very similar to our earlier TCAT study [71, 70, 69]. As described in those studies, the scenarios are inspired from the RoboCup Rescue Simulation domain [50] where heterogeneous groups of agents attend to victims, fires and roadblocks in the aftermath of an earthquake in an urban environment. In our scenarios, a fixed number of robots (homogeneous and
single-task (ST)) and tasks (single-robot (SR) and multi-robot (MR)) are scattered in an office-like environment.

The tasks can be any of the following:

- a *sensor-sweep* task, where a robot is expected to go to a specific location and send back sensor information (e.g., camera image);

- a *fire-extinguishing* response task, resulting from a fire in the environment; and

- a *debris-removal* response task, resulting from structural collapses that create debris.

These latter two types of tasks may require multiple robots to execute, and the fires and debris may block access to areas adjacent to those in which they appear. Sensor-sweep tasks can be executed by a single robot and do not block access.

Different from the TCAT experiment, in this work, the environment is *dynamic*. Human subjects are responsible for revising plans as they execute—i.e., as assigned tasks are completed and additional tasks are introduced. For the experiments, each scenario contains 8 tasks, two of which are available initially. Every 45 seconds, two new tasks are introduced to the environment. In every task pair that is introduced, one of the tasks is an SR task while the other is an MR task requiring 2 robots.

To control for order effect and learning effect, the scenarios were divided into two groups. Human subjects were also divided into two groups, A and B. Scenarios 0-4 are presented for training purposes. Scenarios 5-8 and 9-12 were presented to each participant group using a different interaction mode (manual or collaborative). In both groups, scenarios contained an increasing number of blocking tasks, which are tasks that block access points in the map, such as room entrances, therefore implicitly creating dependencies among tasks (see Chapter 3.2). The exact number and properties of tasks in each scenario can be seen in Table 4.2. The total robot requirement for all the tasks in a scenario is kept constant and is equal to 12 for every scenario ($\sum_{i=0}^{N} req_i = 12$).
### 4.4.2 Procedure

Each experiment started with a training session consisting of five scenarios, labeled 0-4. The first 3 of the training scenarios introduced the TASC user interface and system functionalities to the users, and the last 2 scenarios were mock scenarios to prepare the users to the real experiment.

The training session was followed by 8 experimental scenarios interleaved with surveys about the scenario that had just been completed. Human subjects in Group A completed scenarios 5-8 in manual mode, where all assignments are made manually by the operator and scenarios 9-12 in collaborative mode, where users had the choice of using the “auto assign” button which would assign robots to all unassigned tasks, without changing the existing assignments. Human subjects in Group B completed scenarios 5-8 in collaborative allocation mode and scenarios 9-12 in manual mode. To counterbalance the learning effect, scenarios were randomly picked (from within the groupings listed above) and presented such that the interaction mode alternated between two consecutive scenarios. Upon completion of the scenarios, a post interview was conducted with each participant. During the interview sessions, participants were asked to comment on the interface, scenario features that complicated the planning, general strategies that they employed, and finally the auto assign functionality.

Participants were instructed:

- to distribute tasks to the robots in such a way that the execution of the plan would result in the fastest completion time of all the tasks,
- to maintain a full assignment of tasks at all times, meaning that they should keep adding

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>#SR tasks</th>
<th></th>
<th># MR tasks</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>blocking</td>
<td>non-blocking</td>
<td>blocking</td>
<td>non-blocking</td>
</tr>
<tr>
<td>0-4 (training)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5,9</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>6,10</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7,11</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8,12</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.2: TASC Scenario Properties
tasks to the robots’ task queue, without waiting for the robots to complete their immediate tasks, and

• not to assume that the auto assign button would give the optimal solution; rather it was up to them to decide the performance of the auto-assign tool, as well as when and how to use it, after trying at least a few times.

4.4.3 Metrics and Variables

The analysis of the TASC experiment is conducted in two ways: TAG-level, which is presented here and mission-level, which I’ll cover in Chapter 5.1.

For TAG-level analysis, the dependent variables are as follows:

• **Plan Construction Time** \( (pct) \): elapsed time between the arrival of a new TAG and the time when all available tasks are fully assigned

• **Objective Performance Score** \( (P_{score}) \):

\[
P_{score} = \frac{(ect_{auto} + pct_{auto}) - (ect + pct)}{ect_{auto} + pct_{auto}} \times 100
\]

where \( ect_{auto} \) is the expected completion time of the plan generated by the autonomous planner, \( ect \) is the expected completion time of the assignment generated by the user and \( pct \) is the plan construction time. The addition of the \( pct \) to the \( ect \) is due to the fact that the \( ect_{auto} \) produces a plan as soon as a new TAG arrives. The operator’s \( ect \) is computed when the user assigns all tasks fully. Since, the robots will start to move towards their assigned tasks before a full assignment is reached, correction was necessary for valid comparison of the auto generated plan and the users’ plans. In the above equation, \( pct_{auto} \) is negligible.

As in the case of TCAT experiment, participants provided subjective ratings for mental demand and performance using NASA-TLX survey. Since each scenario contained four TAGs,
there is no exact way of knowing which particular condition contributed more to these ratings. In an attempt to follow the analysis performed in TCAT, in Chapter 3.5, the effects of the independent variables on the subjective NASA-TLX ratings, mental demand and performance, are analyzed.

As in the TCAT experiment, these values are adjusted, as described in Chapter 3.5.3, by shifting each score so that the median rating for that subject was equal to 50. This adjustment was done in order to normalize the ratings across users. The detailed rationale of this process was explained in Chapter 3.5.3.

As stated, mental demand and performance ratings were provided at the end of a scenario. In order to associate these ratings with each of the TAGs in a scenario, mental demand was weighted with the \( pct \) and performance was weighted with \( P_{score} \) values for that TAG. The exact process for mental demand is:

\[
MD_{wjik} = MD_{ij} \times \frac{pct_{ijk}}{\sum_{k=1}^{4} pct_{ijk}}
\]

where \( MD_{ij} \) represents the subjective mental demand rating of participant \( i \) for scenario \( j \), \( pct_{ijk} \) represents the Plan Construction Time of participant \( i \) for scenario \( j \) and TAG \( k \), and the \( MD_{wjik} \) represents the weighted mental demand rating, for TAG \( k \). Similarly for performance:

\[
P_{wjik} = P_{ij} \times \frac{(P_{score})_{ijk}}{\sum_{k=1}^{4} (P_{score})_{ijk}}
\]

where \( P_{ij} \) represents the subjective performance rating of participant \( i \) for scenario \( j \), \( (P_{score})_{ijk} \) represents the Objective Performance Score of participant \( i \) for scenario \( j \) and TAG \( k \), and the \( P_{wjik} \) represents the weighted performance rating, for TAG \( k \).

### 4.4.4 Results and Discussion

The TAG-level analysis is based on the complete and full assignment of tasks manually allocated by the users. However, not all tasks were successfully assigned before the arrival of new TAGs,
resulting in incomplete plans. In order to reduce the number of incomplete plans, the time between the consecutive arrival of TAGs was set to 45 seconds and the number of tasks that were introduced in each TAG was limited to 2. As a result, most participants were able to maintain complete assignments throughout the duration of the mission. Some, however, were not able to do so. An overview of the failed assignments and their distribution over each TAG instance is presented in Table 4.3.

<table>
<thead>
<tr>
<th>TAG ID</th>
<th>#</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.65</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>4.56</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>8.26</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>Total</td>
<td>62</td>
<td>13.48</td>
</tr>
</tbody>
</table>

Table 4.3: Incomplete Plans distributed by TAG ID

As can be seen from the Table 4.3, overall in 13.48% of the TAGs, participants failed to reach full assignment. Most of these failed attempts happened in the third TAG, followed by the second TAG for the duration of the missions. Since the fourth TAG does not have any time limit, there are no failed attempts as expected. These failed runs are removed from the objective workload and objective performance analysis performed in this section.

For statistical analysis of TAG-level data, we used 3-way Repeated Measures ANOVA with independent variables $TDR$ (low, high), $CTR$ (low, high) and $ADD$ (low, high). The data for these independent variables was categorized as low or high by computing 2 clusters using the Expectation-Maximization (EM) algorithm and using the cluster means as a threshold.

All independent variables, $TDR$, $CTR$ and $ADD$ were found to have a significant effect on the Plan Construction Time. For $TDR$, $F(1, 23) = 12.02$ and $p = 0.0023$; for $CTR$, $F(1, 24) = 15.59$ and $p = 0.001$; for $ADD$ $F(1, 25) = 8.39$ and $p = 0.0075$. The box plot of this analysis can be seen in Figure 4.7. There is no significant interaction found among the variables.

None of the independent variables, $TDR$, $CTR$ and $ADD$, however, were found to have any
Figure 4.7: *Plan Construction Time vs. Scenario categories based on TDR, CTR and ADD*

Figure 4.8: *Objective Performance Score vs. Scenario categories based on TDR, CTR and ADD metrics*
significant effect on the Objective Performance Score. For TDR, \( F(1, 23) = 0.035 \) and \( p = 0.85 \); for CTR, \( F(1, 24) = 1.267 \) and \( p = 0.27 \); for ADD \( F(1, 25) = 3.31 \) and \( p = 0.08 \). The box plot of this analysis can be seen in Figure 4.8. There was, however, a significant interaction found between TDR and CTR, \( F(1, 12) = 16.53 \) and \( p = 0.0015 \).

As in the case of objective workload, \( pct \), all independent variables, TDR, CTR and ADD were found to have a significant effect on subjective workload. For TDR, \( F(1, 23) = 21.6 \) and \( p < 0.001 \); for CTR, \( F(1, 24) = 26.9 \) and \( p < 0.001 \); for ADD \( F(1, 25) = 20.501 \) and \( p < 0.001 \). The box plot of this analysis can be seen in Figure 4.9. There were no significant interactions found among the variables.

As in the case of Objective Performance Score, none of the independent variables, TDR, CTR and ADD, were found to have any significant effect on the subjective performance. For TDR, \( F(1, 23) = 3.097 \) and \( p = 0.0918 \); for CTR, \( F(1, 24) = 2.716 \) and \( p = 0.11241 \); for ADD \( F(1, 25) = 0.158 \) and \( p = 0.694 \). There were no significant interactions found among independent variables. The box plot of this analysis can be seen in Figure 4.10.

Summary statistics for objective workload (\( pct \)), objective performance (\( P_{\text{score}} \)), subjective
Figure 4.10: Performance weighted by Objective Performance Score vs. Scenario categories based on TDR, CTR and ADD

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(high, high, high)</td>
<td>NA</td>
<td>18.05</td>
<td>NA</td>
<td>4.52</td>
</tr>
<tr>
<td>(low, high, high)</td>
<td>22.76</td>
<td>2.27</td>
<td>21.71</td>
<td>12.52</td>
</tr>
<tr>
<td>(high, low, high)</td>
<td>43.26</td>
<td>2.96</td>
<td>25.86</td>
<td>12.78</td>
</tr>
<tr>
<td>(low, low, high)</td>
<td>38.64</td>
<td>-5.49</td>
<td>37.0</td>
<td>12.72</td>
</tr>
<tr>
<td>(high, high, low)</td>
<td>50.07</td>
<td>3.52</td>
<td>12.32</td>
<td>12.72</td>
</tr>
<tr>
<td>(low, high, low)</td>
<td>33.72</td>
<td>-5.09</td>
<td>23.5</td>
<td>11.72</td>
</tr>
<tr>
<td>(high, low, low)</td>
<td>70.3</td>
<td>-11.98</td>
<td>35.62</td>
<td>11.49</td>
</tr>
<tr>
<td>(low, low, low)</td>
<td>49.36</td>
<td>-6.0</td>
<td>30.64</td>
<td>12.74</td>
</tr>
</tbody>
</table>

Table 4.4: TASC Summary Statistics
workload (weighted NASA-TLX Mental Demand) and subjective performance (weighted NASA-TLX Performance) are shown in Table 4.4. As the results indicate, all our TAG-level hypotheses hold regarding Plan Construction Time ($pct$), a proxy to objective workload, and subjective ratings for NASA-TLX Mental demand distributed to each tag by $pct$. On the other hand, all of our hypotheses regarding the Objective Performance Score ($P_{score}$) and the subjective performance ratings, NASA-TLX Performance distributed to each tag by $P_{score}$, were invalidated and the only significant effect on the metric was found to be the interaction of $TDR$ and $CTR$. These results can be seen from the boxplots displayed in Figures 4.8 and 4.10. As can be seen in these boxplots, the group means are very close to one another with the exception of the group labeled, $\langle \text{high, high, high} \rangle$, which contains only a single sample. My conjecture for this situation is that the scenarios designed for this experiment were not hard enough to observe differences in performance. I’ll present the explanations regarding this result in the next chapter, in the beginning of Chapter 5.1.1. Also, these results showed that the subjective ratings for workload and performance are consistent with the objective results.

![Figure 4.11: Plan Construction Times vs. TAG ID](chart)

Further analysis were conducted based on the TAG ID’s, to represent the situation at a particular point in time during the experiments. The Plan Construction Time means are displayed
in Figure 4.11, grouped by TAG ID’s. As expected the Plan Construction Times steadily increase as the mission becomes more complex, from TAG1 through TAG4. An important thing to note is that in the first three TAGs, the Plan Construction Times are limited to 45 seconds, which is the time the operators have before the next set of tasks arrive. Since there is no inherent time limit on the last TAG, TAG4, the Plan Construction Times are extended beyond 45 seconds. Also, during the last TAG, it was observed that users had a tendency to keep improving the execution plan as it progressed, therefore removing inefficient allocations that were made in earlier TAGs. This resulted in longer planning times.

Figure 4.12: Objective Performance vs. TAG ID

The objective performance scores grouped by the TAG ID’s are presented in Figure 4.12, which reveals an interesting pattern. On average, the worst performance is displayed by users during the first TAG, where the problem is easiest. Furthermore, the average performance of the users starts to increase as more tasks, therefore new TAGs, arrive. This pattern is broken on the last TAG, where the performance declines. One possible explanation to this situation was provided by the participants during post-experiment interview sessions of the TCAT experiment. One of the main complaints about the TCAT interface was that the estimation of the task execution times using the path lengths was difficult. To remedy this issue, some users suggested extensions to the interface
similar to the timeline component of the TASC interface (such a component was intentionally left out of the TCAT interface, in order to observe human performance without such aids). Based on these statements, it is possible that these distances between the robots and tasks may have been misjudged by users. Even if a better solution would be realized, the users may not have bothered to improve the plan, since the gain would be minimal due to the relatively short cumulative paths that need to be traveled by the robots in the beginning of a mission. As the accumulated total distances become longer, users most likely pay closer attention, due to increased potential gain in good allocation ordering, and start to outperform the greedy allocation strategy. This continues until the last TAG, where the problem becomes complex enough that users’ performance start to degrade.

![Box plot](image)

**Figure 4.13: Mental Demand weighted by Plan Construction Time vs. TAG ID**

The distribution of subjective workload ratings (NASA-TLX mental demand), weighted by the Plan Construction Time (\(pct\)) is shown in Figure 4.13. Similarly, the distribution of subjective performance ratings (NASA-TLX performance), weighted by the Objective Performance Score (\(P_{score}\)) is shown in Figure 4.14.

The summary statistics for these analysis were displayed in Table 4.5.

In order to check the results found in the Three-Way Repeated Measures ANOVA analysis,
Figure 4.14: *Performance weighted by Objective Performance Score vs. TAG ID*

Table 4.5: *TASC Summary Statistics grouped by TAG ID*
similar analysis to the one done in Chapter 3.5.4 was attempted. For this purpose similar to the independent variable Quadrant, Octile variable is created for the 8 groups \((TDR \times CTR \times ADD)\), however the assumption of sphericity is violated, therefore One-Way Repeated Measures ANOVA analysis is not used for the TASC experiment. Also, similar to the TCAT, post-hoc analysis was not needed since all independent variables have 2 factors, high and low.

4.5 Summary

In the previous chapter (3), I introduced a graph-based model, TAG, that allowed us to represent task scheduling problems from a human’s perspective, and at an abstract level. Then through a user study, we verified the model’s two key metrics (APR and CTR) as factors that have a direct effect on mental workload and perceived performance in static domains. In this chapter (4), first, I presented an extension of the TAG model, the MAP, as a sequence of TAGs, in order to represent dynamic environments with changing characteristics. In this extended model, three metrics were investigated: CTR, TDR and ADD. Since APR was clearly a dominant factor in the TCAT study, it was omitted from the TASC experiment, therefore scenarios were designed to introduce tasks by keeping the APR constant. CTR was included because in static environments its effect was surprisingly not strong. I further wanted to investigate its effects in dynamic situations. While ADD accounts for the number of possible assignments in a problem instance, TDR was investigated as the only metric that represents the dynamic nature of the problem domain.

Following from the TCAT experiment, a new software tool, the TASC interface, was designed and developed. This consisted of an interface, an automated planner and protocols to communicate with autonomous robot controllers, operating in a simulated environment (Stage). The TASC tool allows its users to supervise a team of robots in two interaction modes: manual allocation and collaborative allocation, where the users may choose to relinquish the assignment of tasks to the autonomous planner. The implementation details and the functionality of the tool
was described in this chapter.

Then, the metrics that are derived from the model were verified through a user study. The analysis and the results of the data at each decision point, which is referred to as TAG-level, was investigated and all metrics were found to have a significant main effect on the objective workload metric, Plan Construction Time and subjective NASA-TLX mental demand ratings. Contrary to our hypothesis, however, the factors that these metrics are representing, did not have any effect on the objective performance metric, $P_{score}$ and the subjective NASA-TLX performance ratings. I hypothesize that this latter result is due to the relative ease of the scenarios, as average ratings of the participants suggest (see Chapter 5.1.1), designed to increase the number of complete plans generated during the mission. Regardless of the underlying reason, the investigated factors, constraint property, spatial distribution of robots and the interruption of the operator due to the dynamic nature of the domain, were verified to have an effect on the Plan Construction Time, which is a proxy for mental workload. This analysis directly answers the research question, RQ4, posited in Chapter 1.2, for these investigated factors.

In the next chapter, I’ll continue with the analysis of the TASC experiment, this time focusing on the mission-level data, as well as subjective ratings and surveys conducted during the experiment.
Chapter 5

Mission-level Analysis and Integration of MAP model into MRTA methods

In Chapters 3 and 4, I have described a model and verified it in first static domains and observed human operators’ strategies and behavior in a stress-free environment. Then, the same model is extended and analyzed in a dynamic, real-time environment. Although the overall problem was the same, during the experiments, we observed a drastic but not surprising change in the human operators’ strategies. While in the TCAT experiment, the operators were careful and reported to calculate several alternatives before finalizing their plan, in the TASC experiment, due to time pressure, this behavior shifted towards greedy allocation of robots to tasks. This shift in operator strategy is rational since the robots started moving towards their immediate targets as soon as they were assigned. Therefore operators were able to “buy time” for assigning remaining tasks, once each robot is assigned at least one task to execute.

Also, in Chapter 4, the interaction between the TASC interface and the operator was studied in two modes: manual and collaborative.

In these chapters, research questions; RQ3 (What are the factors and to what degree they affect human supervisor’s decision making regarding task assignment under stress-free and favorable
circumstances?) and RQ4 (What are the factors and to what degree they affect human supervisor’s decision making regarding task assignment during real-time high-stress missions?) were addressed and the extended TAG and MAP models were verified via empirical means. I will now address research questions RQ5 (What are the common strategies adopted by the supervisors?) and RQ6 (How can useful human strategies be integrated with common MRTA techniques?).

After each scenario, participants were subjected to the NASA-TLX survey and a survey about the automated planner, if they used it for a given scenario. At the end of each experiment, participants were asked several questions in an interview session, some directed towards RQ5 and RQ6.

In the remainder of this chapter, I first present the results of the analysis for mission-level data of the TASC experiments. Next, the answers provided to the post-experiment interview questions are discussed in detail. A review of common heuristics devised for MRTA is presented. Finally, the strategies employed by the participants and their integration into common MRTA algorithms, namely auction mechanisms and Constraint Programming (CP) techniques, as explained in Chapter 2.2, are discussed.

5.1 Mission-level analysis of TASC experiment

The second goal of the TASC experiment, as mentioned in Chapter 4.4, was to observe general human interaction with the autonomous planner and the effectiveness of the collaborative approach. The results and analysis were deferred to this chapter. The analysis of this interaction between the human operator and the autonomous planner helps us to answer research questions R5 and R6, which in turn address the main questions R1 and R2, as specified in Chapter 1.2. In order to address these questions, I am specifically interested in observing:

i. the degree of trust placed by users in autonomously generated solutions,
ii. how users are utilizing this auto-complete tool, and

iii. how well they can assess the quality of autonomously generated solutions.

The general hypothesis regarding the use of autonomous planners is that, with the right interaction scheme, autonomous planners can reduce the mental workload of human operators, thus leading to higher situational awareness and performance. In the TASC experiment, the autonomous planner was not designed to provide a sophisticated, optimal plan, but rather a greedy approach, as described in Chapter 4.2.2, which performed reasonably well and was used as a baseline. The rationale for this decision is to allow the participants to decide the quality of the solution and how to use the tool, in accordance with item (i) in the previous paragraph.

The hypotheses for the TASC experiment in regard to the overall mission is as follows:

1. The mental demand for human subjects who use “collaborative allocation” mode is lower than the mental demand for human subjects who use “manual allocation” mode.

2. The performance of human subjects who use “collaborative allocation” mode is higher than the performance of human subjects who use “manual allocation” mode.

For mission-level analysis, data was divided according to interaction mode for each scenario. To test my mission-level hypotheses, I used both objective and subjective metrics as dependent variables in my analysis. The objective metrics are:

- **Run time**: The amount of time between the beginning of the scenario (appearance of the first TAG) and the completion time of the last task in the scenario;

- **Activity time**: The total amount of time spent actively by the user to add/delete task assignments;

- **Interaction time**: The amount of time between the first assignment and the last assignment;
• **Final observation time:** The amount of time between the last assignment and the completion time of the last task in the scenario; and

• **Total deliberation time:** Calculated as, \( \text{run time} - (\text{activity time} + \text{final observation time}) \).

While the run time of the mission is used as the objective performance metric, the remaining metrics, activity time, interaction time, final observation time and total deliberation time are related to the percent busy time, which is a valid proxy for mental workload as discussed in Chapter 2.3. For subjective metrics, I used the NASA-TLX survey [1], conducted at the end of each scenario.

### 5.1.1 Analysis of Mission-level Objective Metrics

An overview of the objective metric comparisons between the manual allocation and collaborative allocation mode groups can be seen in Figure 5.1. Overall, the main objective performance parameter, run time (Figure 5.1a), does not seem to be different between the two groups. This result is also confirmed with further analysis presented at the end of this section. This outcome is a direct result of the experiment scenario design decision and the auto assign algorithm efficiency. In order to ensure that enough data could be collected for the verification of the model, participants needed to be able to present completed task plans between TAGs. This required providing ample time between TAGs and limiting the total number of tasks, therefore simplifying the scenarios. In addition, in order to observe the full range of user trust on the automated planner, its algorithm was kept as a baseline, greedy allocation scheme, which is sub-optimal.
For the remainder of the metrics, interaction, activity and the last observation times seem to be favoring the collaborative allocation group, which means that the same performance in terms
of mission run time can be achieved with less interaction and activity in collaborative allocation mode than in manual allocation mode. The total deliberation time also seems to be better for the collaborative allocation group although the difference is not as pronounced as the other three metrics.

The mean and the standard deviation of the objective metrics are presented in Table 5.1.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Manual (MAN)</th>
<th>Collaborative (REL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run-time (sec)</td>
<td>332.5</td>
<td>323.4</td>
</tr>
<tr>
<td>Interaction time / Run-time * 100</td>
<td>64.8</td>
<td>57.9</td>
</tr>
<tr>
<td>Activity time / Run-time * 100</td>
<td>14.1</td>
<td>9.2</td>
</tr>
<tr>
<td>Total Deliberation time / Run-time * 100</td>
<td>52.6</td>
<td>50.2</td>
</tr>
<tr>
<td>Final Observation Time / Run-time * 100</td>
<td>33.3</td>
<td>40.7</td>
</tr>
</tbody>
</table>

Table 5.1: Mission-level Time metrics

<table>
<thead>
<tr>
<th>objective metrics</th>
<th>U</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>run-time</td>
<td>6532</td>
<td>= 0.504</td>
</tr>
<tr>
<td>interaction time (%)</td>
<td>8321</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>activity time (%)</td>
<td>9359</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>total deliberation time (%)</td>
<td>7200</td>
<td>= 0.039</td>
</tr>
<tr>
<td>final observation time (%)</td>
<td>4011</td>
<td>&lt; 0.001</td>
</tr>
</tbody>
</table>

Table 5.2: Mann-Whitney-U test results for objective metrics

For statistical analysis of the objective metrics, the Mann-Whitney U test has been used, because the data did not meet the assumption of normality. The differences between the groups for the objective metrics, along with their significance, are displayed in Table 5.2. This data confirms the results displayed in Figure 5.1 and the Table 5.1. According to these results, the difference between the group means for run time is indeed non-significant. However for all other metrics (interaction, activity, final observation and total deliberation times) the difference is significant. It’s worth noting that total deliberation time means for the groups is only marginally significant (0.05 > p > 0.001).
5.1.2 Analysis of Mission-level Subjective Metrics

As stated in Chapter 4.3, each scenario was followed by a NASA-TLX survey, and if the interaction mode was collaborative, then another survey for the subjective assessment of the auto assign tool was also presented. In this section, the NASA-TLX survey data analysis is presented.

The overview of the NASA-TLX dimensions and the final score is presented in Figure 5.3, grouped by interaction mode. The only NASA-TLX factor that is not included in the analysis is physical demand, since the task does not require any physical activity. According to the figures, all factors except performance seem to be different between the groups. The subjective performance, similar to its objective counterpart seems to be the same in both groups, suggesting that users have accurately assessed their performance. Among the other parameters, the most important one is mental demand since it is directly the metric for one of the mission-level hypotheses. According to these figures, mental demand, temporal demand, effort, frustration and the overall NASA-TLX score is lower for the collaborative allocation mode group. The descriptive statistics are presented in Table 5.3.

To test if the differences in means, that are observed in these figures, are significant further statistical analysis have been conducted. For this purpose, the Mann-Whitney U test has been used, since the data did not meet the assumption of normality. The results of this analysis is presented in Table 5.4.

<table>
<thead>
<tr>
<th>NASA-TLX Metric</th>
<th>Manual (MAN)</th>
<th>Collaborative (REL)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>µ</td>
<td>σ</td>
</tr>
<tr>
<td>Mental Demand</td>
<td>47.9</td>
<td>23.5</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>45.9</td>
<td>24.9</td>
</tr>
<tr>
<td>Performance</td>
<td>65.4</td>
<td>20.3</td>
</tr>
<tr>
<td>Effort</td>
<td>47.6</td>
<td>23.8</td>
</tr>
<tr>
<td>Frustration</td>
<td>33.1</td>
<td>25.8</td>
</tr>
<tr>
<td>NASA-TLX Score</td>
<td>52.3</td>
<td>15.6</td>
</tr>
</tbody>
</table>

Table 5.3: NASA-TLX metrics summary

According to these results, mental demand and overall NASA-TLX score means are significantly different between the groups. Also, temporal demand, effort and frustration means are...
(a) Mental Demand

(b) Temporal Demand

(c) Performance

(d) Effort

(e) Frustration

(f) NASA-TLX Score

Figure 5.2: NASA-TLX Metrics displayed by interaction mode
marginally significantly (0.05 > p > 0.001) different between the groups. The subjective performance means, as Figure 5.2c shows, do not differ significantly between the groups.

<table>
<thead>
<tr>
<th>subjective metrics</th>
<th>U</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>mental demand</td>
<td>7479</td>
<td>= 0.008</td>
</tr>
<tr>
<td>temporal demand</td>
<td>7294</td>
<td>= 0.024</td>
</tr>
<tr>
<td>performance</td>
<td>6347</td>
<td>= 0.776</td>
</tr>
<tr>
<td>effort</td>
<td>7243</td>
<td>= 0.032</td>
</tr>
<tr>
<td>frustration</td>
<td>7168</td>
<td>= 0.047</td>
</tr>
<tr>
<td>NASA-TLX score</td>
<td>7491</td>
<td>= 0.0078</td>
</tr>
</tbody>
</table>

Table 5.4: Mann-Whitney-U test results for subjective metrics (NASA-TLX)

5.1.3 Analysis of Interaction and Plan Execution Data

In addition to deletion of assignments, the TASC interface provides additional features and ways to modify the plans as described in Chapter 4.2. An overview of these functionalities and the modifications are presented in Table 5.5. As can be seen, the modification to the plans were mainly done by deleting individual assignments, which on average is done 4.56 times per scenario for manual allocation and 5.05 times per scenario for collaborative allocation groups, out of 12 total assignments per scenario, which is not a significant difference. Clearing a robot’s agenda or a task’s assignment is done by selecting the object and hitting the DELETE key, which seems to be done rarely for both interaction modes. The “Clear” button also seems to be rarely used, however in the collaborative allocation group, on average, this button was used twice as often than it was used by the manual allocation group. The “Wait” button, intended as a panic button, was never used by the manual allocation group and very rarely by the collaborative allocation group.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Manual (MAN)</th>
<th>Collaborative (REL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. # Delete Assignment</td>
<td>4.56</td>
<td>5.05</td>
</tr>
<tr>
<td>Avg. # Clear Robot’s Agenda</td>
<td>0.35</td>
<td>0.26</td>
</tr>
<tr>
<td>Avg. # Clear Task’s Assignments</td>
<td>0.15</td>
<td>0.08</td>
</tr>
<tr>
<td>Avg. # Clear Button Clicked</td>
<td>0.14</td>
<td>0.28</td>
</tr>
<tr>
<td>Avg. # Wait Button Clicked</td>
<td>0</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 5.5: Mission-level Interaction metrics: Button-clicks and Keystrokes
Aside from the run time of the mission, data about other metrics related to the executed plans have been collected:

- **Total path length**: Cumulative path length traveled by all the robots during the mission.

- **Max path length**: Cumulative path length of the robot, that traveled the furthest during the mission.

- **Total idle time**: Cumulative time where each robot is not assigned to any task

- **Max idle time**: Cumulative idle time of the robot which waited unassigned the longest during the mission.

- **Total delay time**: Cumulative time where robots stop deliberately to avoid collision.

- **Max delay time**: Cumulative delay time of the robot which waited the longest for the other robots to move out of its path, for the duration of the mission.

Although these metrics were not critical from the point of view of this work, they appear commonly as task effectiveness (TE) metrics in MRS and MAS literature. A summary of descriptive statistics for these metrics is presented in Table 5.6. One of these metrics is important for the assessment of run time between manual and collaborative allocation groups: the **total delay time**. As explained above, this value is the direct result of the collision avoidance protocol explained in Chapter 4.3, therefore reflects the amount of congestion occurring as a result of task assignments. Since users have limited control over the congestion and detours, and since total delay time is included in mission run time, the run time analysis would require adjustment for accurate assessment, if these values were significantly different between the two groups. As seen in Table 5.6, mean and standard deviation of total delay time for both groups is relatively similar and negligible compared to the run times, constituting < 1% of the overall mission.
5.1.4 Analysis of Collaborative Control Functionality

One goal of the TASC experiment was to observe the interaction between the operators and the autonomous planner, which was implemented using a sequential greedy approach, explained in detail in Chapter 4.2.2. The users were asked to use it any way they saw fit and at the end of each scenario, they were asked to rate:

1. usefulness of the functionality,

2. quality of the solutions produced by the tool, and

3. confidence in the accuracy of the ratings they provided.

The results of these ratings are presented in Figure 5.3. In the first two figures, 5.3a and 5.3b, it can be clearly seen that the majority of the participants rate the usefulness and the solution quality above 50%. However, a group of the participants rate both the usefulness and the quality of the solution, on the lower end of the rating scale, among which a number of them found the auto assign functionality completely useless. The importance of this result becomes more apparent when these ratings are adjusted by the confidence of the users, as can be seen in Figures 5.3c and 5.3d. In these Figures, the low ratings are more pronounced, signifying that the group of participants who did not find the auto assign tool useful at all are strongly opinionated on the matter. Also, there is a shift in the overall adjusted ratings towards lower values, suggesting that the participants who rated high are less confident about both the quality and the usefulness of the tool. The possible
explanations for these results will be covered in Chapter 5.2.3 where I discuss the findings of the post-experiment interviews.

Further analysis was conducted based on some objective metrics related to the usage of the auto assign functionality. In order to assess the operator’s dependence on the auto assign tool, the percentage of tasks assigned by the auto assign button was logged. The results are presented in Figure 5.4. In the first four sub-figures, percentage of assignments made by the tool is presented during each TAG. $TAG_1$ represents the situation when the scenario starts. From Figure 5.4a, we can infer that users either chose to start the assignment with auto assign or, as most of them did, assign all tasks manually. This manual assignment preference starts to change as missions progress.
The usage of the auto assign functionality increases, in terms of the percentage of tasks assigned, in Figures 5.4b through 5.4c and finally becomes the dominant method of assignment in Figure 5.4d. The percentage of tasks assigned for the entire duration of the mission is shown in Figure 5.4e.

5.1.5 Summary

As stated in Chapter 5.1.1, the objective performance measure, run-time, and in Chapter 5.1.2 the perceived performance values from NASA-TLX survey consistently state that there is no significant difference between manual allocation and collaborative allocation modes. These results show that the null hypothesis cannot be rejected; using collaborative allocation does not lead to better performance, for the second mission-level hypothesis. This holds true for both objective performance (run time) and subjective performance acquired from the NASA-TLX.

The subjective mental demand metric is significantly different between the group using the collaborative allocation and the group using manual allocation mode, therefore my first hypothesis holds for subjective mental demand. At the mission-level, there is no objective metric for mental demand, however, interaction time, activity time and observation time values directly influence operator free time, which leads to higher situational awareness. Collaborative allocation interaction time and activity time compared to manual allocation interaction time and activity time shows that collaborative allocation reduces time spent interacting with the GUI, giving more free time to the operator, without sacrificing performance. This observation is also supported by the subjective metrics for collaborative allocation vs. manual allocation: temporal demand and effort.

In Chapter 5.1.4, the data for the usefulness and the quality of the solution produced by the auto assign tool is presented. The results shown that although the usefulness and the quality of the solutions are rated high by the majority of participants, when these values were adjusted by confidence values, it was seen that participants who rated the tool’s qualities high did not seem to be confident about their answers. These results may be a sign for automation bias, where users
Figure 5.4: Tasks assigned by Auto Assign Functionality
have excessive trust on the system’s capabilities, based on their experiences during prior scenarios, however, they do not have the full grasp of the tool’s performance in all situations, as some of them stated during interview sessions. In addition, a portion of the population rated the tool as almost useless, and also had high confidence in these ratings. Since there is not a significant difference in objective performance between the collaborative allocation and manual allocation groups, it is reasonable to infer that these ratings are rather biased, and given as a result of frustrating experiences using the tool. Although the reasons for these ratings are not fully known, some are expressed by the participants during the post-experiment interview sessions. It is safe to say that, frustration may stem from a gap between the expectations of the operator and the underlying algorithm behavior, unclear inner workings of the automated system and the lack of mechanisms for interaction to adjusting the behavior of the system. The devised models and the research questions addressed in this dissertation aim precisely to remedy these situations, therefore improve the team’s performance.

5.2 Post-experiment interviews

In addition to the results analyzed in the previous section, the participants were interviewed after the experiment. In this interview, four questions were asked in an informal setting:

1. How was your experience with the tool? Was it easy or hard to work with? Any comments about your experience?

2. After going through all the different scenario types, which ones do you think stand out as harder ones?

3. Did you have a particular strategy for allocating the tasks? When you encountered a scenario for the first time, did you always start allocating the same way, or did you have several different strategies for different situations?
4. How was your experience with the auto assign feature? What were the particular situations (if any) when you felt the need for support, therefore left the decision to the auto assign functionality? What was your strategy for using the auto assign functionality?

The answers to these questions, provided by the participants, are consistent with my analysis and revealed much insight about users’ approaches and strategies for the supervision of a multi-robot team. I’ll discuss how to integrate these strategies into the commonly employed MRTA techniques at length in Chapter 5.3, in the context of designing automated task planners.

5.2.1 Usability and Task Complexity

First I’ll review some reported issues regarding the GUI and the system, which is required for correct assessment of the user responses. For the TCAT experiment, when participants were asked to comment about the GUI, all responded that it was simple, easy to use and worked well for the task at hand. One comment that was made by most participants was that it would have been better to be able to do the assignments on the map component rather than the allocation component. The TASC interface was kept simple and consistent with the interaction scheme introduced in TCAT. In the TASC experiment, although all users except one were generally content with the functionality of the interface, there were a number of issues reported by users:

- overlapping paths were hard to distinguish
- task and robot selection and deselection was confusing
- allocation should be done in map component
- auto-complete prioritization wasn’t clear
- ordering of the allocation wasn’t clear
- fonts on the robot and task labels were small and hard to read when they overlap
• assigned robots should move closer to their targets even when their task is blocked

and some suggestions to improve the interface:

• distinguish paths and robots with different colors
• timeline should show the expected time of each action wait, move and task execution
• allow selection of robots and tasks with keyboard hotkeys
• comparison of existing and modified plan assignment would be helpful.
• an undo functionality to remove the last assignment rather than delete
• mouse right-click to delete

Judging by the responses given in the NASA-TLX components, with respect to effort (Figure 5.2d, which represents in this context the usefulness of the GUI and the interaction scheme) and frustration (Figure 5.2e, which reflects the overall experience of the users), it is reasonable to assume that these issues did not affect the results in a major way and that the TASC interface worked sufficiently well for the presented scenarios.

When asked about the properties that make scenarios hard, in the TCAT experiment, the responses varied but commonly converged on two properties: number of multi-robot tasks and the number of good alternative assignment choices. These two properties also held true in the TASC experiment, in addition to a number of other properties related to the dynamic nature of the scenarios:

1. Plan disruption
2. Number of multi-robot (MR) tasks
3. Number of blocking tasks
4. Multiple candidate solutions
5. Number of assignments

6. Spatial task distribution

The most commonly reported property that made the task assignment hard was the arrival of a task, which blocked robots’ paths to certain targets, rendering them useless and not navigable until the newly arrived task is scheduled to be cleared. This is stated as Plan disruption in item (1). The TASC system, by default, removed such invalid paths and all paths that come after them. Although this feature was necessary, as a result of it, a subset of the assignments were removed instantly, therefore requiring additional cognitive workload for the operator to adjust to the new situation and update the schedule. One participant stated that they started to think towards where such tasks may appear and they disrupted the execution of the current plan, and set alternative courses of action if such an event took place.

Another common answer was the number of multi-robot (MR) tasks in a problem instance, item (2). This property mainly seems to generate additional workload in three ways. First, if the task is a blocking task then it becomes a simultaneous rendez-vous problem, which becomes harder when there are other tasks of the same nature. The second case is related to a contradiction with a strategy that some participants use: team or coalition formation. In this strategy, participants exclusively try to use the same two robots to attend to MR tasks, which always happened to require two robots. In certain situations, when there are a number of MR tasks, it is more beneficial to break up the team and use other robot pairings. Users reported that breaking up a dedicated MR task team takes longer to decide, a bias that arises from committing to the strategy. Lastly, when the MR tasks are spread out, it simply becomes harder to judge the distances between robots to tasks and between tasks. When the tasks are MR, the problem becomes harder. Some users reported sparse spatial distribution of the tasks as the main factor which makes a problem instance hard, as stated in item (6).
The existence of blocking tasks in a situation is also reported as a major factor which makes the assignment more complex, item (3). An instance of this case is when a single-robot (SR) task blocks the entrance to an area with an MR task. The system, with no particular reason or intent, is designed so that the robots only move towards their immediate targets if the path is clear. When two robots, \( r_1 \) and \( r_2 \), are assigned to a MR task, \( t_A \), in a room and \( r_2 \) is assigned to clear the SR task, \( t_B \) that blocks it, only \( r_2 \) moves to clear \( t_B \), while \( r_1 \) waits until \( t_B \) is cleared. This was stated to be a source of frustration when planning for further steps.

The users also stated that multiple good and conflicting alternative assignments, item (4), and the total number of assignments to perform, item (5), made the problem harder, as was also reported in the TCAT experiment.

Finally, some users reported some bias in their decision making. One user stated that the fire tasks seemed like a higher priority and so they tended to think about them first even when they were fully aware of the instructions about all tasks being of equal priority. Two other users stated that when a robot is idle, it caused frustration, and forced them to search for an alternative assignment, even if there was no better alternative in that situation.

5.2.2 Strategies employed in TCAT and TASC for task assignment

The general strategies adopted by the participants in the TCAT experiment were mostly related to prioritizing the solution to certain task properties and then considering other alternatives in an exhaustive manner. This was common for most participants and made sense given that there were no time restrictions. Generally, the strategies reported to be employed by the participants were a mixture of following:

- greedy assignment minimizing travel distance
- single-robot tasks first
- trapping tasks first
• blocking tasks first

• multi-robot tasks first

• allocation based on area/sector of responsibility

All participants stated that they used some subset of these strategies depending on the situation. However, since the TCAT experiment data only shows the end result, we do not have data regarding the plans constructed with particular strategies and then modified to a better alternative. Therefore, accurate interpretation of these statements given by participants is, that these strategies were used at some point during the construction of a task schedule without explicit description of the conditions that they were used.

In the TASC experiments, all of these strategies, except the trapping-task first, were reported to be used by the participants. The main difference between the reports provided in these two experiments is that in TASC, the participants stated that they were very conservative about modifying their initial approach to the problem even if it was realized to be flawed in later stages of planning. This behavior is understandably a result of the time constraint imposed on the participants and the effort required by the interface to modify the plan. In addition, due to the dynamic nature of the scenarios and constrained with time, some participants used strategies which were not seen in the TCAT experiment.

A summary of strategies reported for task assignment is as follows:

• greedy assignment in terms of earliest completion time

• blocking tasks first, (regardless of MR or SR)

• multi-robot tasks first

• single-robot tasks first

• allocation based on area/sector of responsibility
• form sub-teams/coalitions, assign sub-teams to tasks with matching required platform property

• uniform distribution of tasks to robots

• leaving one robot idle for future tasks

• centrally located tasks last

• centrally located blocking tasks first

Participants explained their rationale for using greedy allocation, prioritizing multi-robot tasks, single-robot tasks and blocking tasks as well as allocating robots in the same manner as in the TCAT experiment.

Forming a coalition between two robots and assigning them to only MR tasks, while assigning the other robot to SR tasks was the second-most commonly reported strategy, after greedy assignment. The fact that this was not reported in the TCAT experiment suggests that the time pressure to complete the task assignment may make this strategy appealing. One explanation for this could be that committing the same robots to a particular type of task reduces the possible assignments therefore speeds up planning.

The remaining strategies, uniform task distribution, leaving a robot idle and prioritizing centrally located tasks, are stated to be related to the expected arrival of new tasks. By distributing tasks uniformly, modifications of the plan can be made easier. Leaving one robot idle allows for tending to tasks that appear in places that are far away from the remaining tasks. Participants who reported that they assigned the centrally-located tasks last explained that their rationale was to place the robots at a central location once the plan is complete, therefore they can be assigned to newly arriving tasks and reach them faster, no matter where it appears on the map. On the other hand, if a blocking task appears at a central location, some participants chose to assign them first, since they pose a higher risk of blocking off a bigger portion of the map, making it hard to
reach tasks on the other end.

One participant reported that forming a coalition of two robots and assigning them to MR tasks caused the robots to get in each others’ paths, triggering the collision avoidance protocol, which slowed the travel time of the robots. As an alternative, they chose to assign robots to MR tasks that are located away from each other, thus reducing the congestion. Another participant stated that the disruption of the plan by a newly arriving task, was so frustrating that they started thinking about how to recover from certain imaginary scenarios. Lastly, a participant reported a bias towards assigning tasks in the order that they appeared in the scenario.

5.2.3 Collaborative Allocation Functionality

To the last interview question, regarding the auto assign functionality, most participants responded that they found the auto assign functionality to be useful and to produce good quality solutions. This coincides with the most commonly used allocation strategy, reported by participants (Chapter 5.2.2): greedy allocation. General strategies for using the auto assign tool are stated as:

- use it immediately when a new task appears and then fix if necessary
- use it only as a last resort
- use it after assigning the high priority tasks

When further questioned, the users stated that it was hard to trace the auto-generated plans when there were too many tasks. As a result, some of the users used the auto assign tool and did not check the plan because they trusted it from prior experience (automation bias). Others stated that they did not use the functionality as much as they wanted, or used it in trivial and simple cases, due to the fact that it was hard to understand and assess the plan generated by the tool, therefore they didn’t trust the outcome. Some users suggested displaying the auto-generated plans in iterative steps for ease of understanding.
Some participants stated that the auto assign tool was useless and the solutions were sub-optimal. Follow-up statements from these participants on this matter revealed that they commonly tested the tool’s performance in the beginning and if they found that the solution was not good, they stopped using it completely for the rest of the mission. This user behavior is commonly seen in interaction with almost any system: if users do not like a particular functionality, they stop using it altogether if they can perform the tasks by some other means, as stated in previous works by Kichkaylo et al. [49] and Clare et al. [12]. It was no different for the TASC experiment.

The reasons varied among the users who rated the usefulness of the auto assign feature low, regarding what these users did not like about the auto assign feature. Some attributed the sub-optimal performance to situations where the auto assign was used upon arrival of new tasks and it did not modify the existing assignments, as it was designed, therefore resulting in sub-optimal situations. An example of this was given by one of the participants, where a new task appears next to a robot that was on its way to another task, and when the auto assign button was clicked, it did not change the ordering of the robot’s assignments and send the robot to the newly arrived task first, as the user had expected. This behavior of the auto assign feature was explained during the training sessions. Aside from these few user or training errors, it is reasonable to assume that most of these low ratings are related to the quality of the auto-generated solutions, since it is clear from the results of the TAG-level data analysis, presented in Chapter 4.4.4, that many participants were able to generate plans that outperformed the greedy assignment used by the auto assign tool.

5.3 Integration of TASC heuristics into Autonomous Planners

The utilization of autonomous planners are common in SOMR control schemes as decision support tools, and some examples were covered in Chapter 2. Two of the most commonly used techniques, CP and auction-based mechanisms are discussed in detail in Chapter 2.2. Here, I’ll discuss some of the ways to implement TAG and MAP model parameters as guiding heuristics for these methods.
In addition, some of the commonly used user strategies explained previously, in Chapters 5.2.2 and 5.2.3, and how to incorporate these into the design of autonomous task planners will be discussed.

Use of heuristics for the MRTA problem is not a new concept. General purpose heuristics have been developed for both CP techniques and auction mechanisms. For CP techniques, the search order and value assignments play a huge part in the solution in the case of CSPs and DSCPs and also in speed and performance in the case of COPs and DCOPs. A variety of strategies were tried in previous research and applications in CP field, in order to improve the search. A few are discussed below.

One approach, constraint propagation or consistency algorithms, sought to reduce the variable domain sizes by removing inconsistent value assignments from variables’ domains. The constraint propagation approaches, arc-consistency and path-consistency algorithms, were covered in Chapter 2.2.2.

Aside from constraint propagation algorithms, heuristics have been commonly used to guide search. These heuristics can be categorized into two classes, based on the property of the constraint network that they order:

- **Variable ordering heuristics**

- **Value ordering heuristics**

Variable ordering heuristics are used in sequential-search based algorithms, where each variable is processed one at a time and assigned a value from its domain. Backtracking algorithms are perfect examples for sequential-search based algorithms, where variable ordering heuristics determine the search performance in a significant way. Since, for a CSP or DSCP algorithm, finding a solution means termination of the algorithm, from an HRI perspective, it also means that this solution will be presented to a human operator.

As mentioned in Chapter 2.2.2, one successful heuristic is called *min-conflict* which is used with backtracking. In min-conflict, each variable is given a number representing its consistency
with other variables and when a conflict occurs, the variables with least consistency are changed first, in order to prevent diverging from a possible solution. Another commonly used heuristic is based on the fail-first principle [38], which gives priority to variables which are most likely to fail for value assignment. This way, the search tree is pruned and the possible value assignments that are inconsistent with the prioritized variables are discarded during early stages of search. This also means that the search algorithms detect conflicts and terminate early if the problem has no solution. Determining which variables are likely to fail depends on the context and problem domain. Heuristics such as choosing the variable with the smallest domain first or the one that is most constrained first are examples that follow the fail-first principle. Variable ordering heuristics cannot be used with iterative improvement algorithms, since in these algorithms all variables are assigned a value in parallel.

Value ordering heuristics can be used both in backtracking and iterative improvement type algorithms. According to these heuristics, the values in the variable domains are ordered according to some criteria. This effectively means that algorithms try to assign particular values earlier than others.

The modeling of the dynamic problem environment discussed in this dissertation is a challenging one. Although there are lots of similarities between a TAG and a constraint network, the main issue is related to constant update of the vertex domains in the constraint network. If the problem is modeled as a CSP such that each vertex represents a task, then the initial domain of each vertex has to contain only the robots that can access it at that point in the partial assignment. As robots are assigned to critical tasks however, the domains of the tasks need to be updated, since a blocked robot or task suddenly becomes accessible. In short, the problem investigated in this dissertation has the property that requires constant change in domains of variables, which standard CP algorithms are not designed to handle. One naive approach is to treat each TAG as a sub-problem, and assign all tasks with non-empty domains. Then update the values and domains
of tasks, and repeat this process until all tasks are assigned. Some of the strategies employed in
the TASC experiment can be directly used as a variable ordering heuristic. These strategies are,
as listed in Chapter 5.2.2:

- greedy assignment in terms of earliest completion time
- blocking tasks first, (regardless of MR or SR)
- multi-robot tasks first
- single-robot tasks first
- centrally located tasks last
- centrally located blocking tasks first

Auctions, similar to CP techniques, can be guided by heuristics to divide tasks among
robots. Auction mechanisms are by their nature computationally distributed, therefore the resulting
allocations are generally assessed by factors such as social welfare and pareto optimality. The
resulting allocation of a single-item auction depends on the mechanism’s clearing rules as well as
the bidding strategies of the robots. When allocating multiple items, as covered in Chapter 2.3, how
the auctions are announced, in parallel or sequentially, plays an important role. In the former case,
the robots have to bid for all items at once, therefore clearing rules and bidding strategies are the
only ways to steer the allocation to a desired point. This, of course is valid only in cases where the
bidding rules can be set by the designer, and in team settings. For the latter case, where auctions
are carried out sequentially the bidders can adjust their state as items are allocated, therefore
making more informed bids for the subsequent auctions. In this situation, the order in which the
items are put up for auction becomes an important factor and another way to steer the result of
the allocation.

One possible and straight-forward way to design a steerable autonomous planner is using an
SSI auction, which the current allocation algorithm, used by the TASC, is based on. In this setup,
the above discussed strategies, according to users’ preference, can be used as heuristics to associate a weight to each task, as in the work, by Clare et al. [11], described in Chapter 2.3. Then the tasks can be sorted according to their weights and auctioned one by one, starting with the highest weighted task first.

The SSI auction can be steered in the same manner for the TASC experiment heuristics discussed for the CP model. While in the CP case, the search order for the variables is set by these heuristics, in SSI, the tasks can be auctioned according to the same ordering. The strategy in which the robots are allocated to tasks in certain pre-defined areas or sectors of the geographic map can be implemented in an SSI auction setting in two ways: either regulated by the auctioneer in the clearing rule or by the robots themselves in the bidding rule. In both cases a penalty can be incurred on bids for the tasks that are outside of the robots’ jurisdiction. Also, as a clearing rule, bids can be removed from the domain of task vertices unless they become empty, or as a bidding rule, the robots can simply ignore such tasks.

Implementation of a commonly used strategy, forming sub-teams/coalitions and using them as much as possible, requires additional clarification as to when would it be reasonable to break up the coalition. The answer to this question depends on operator preferences. For now, I’ll assume that a hypothetical threshold, $\alpha$, exists, which can be set by an operator, representing the time that can be gained by breaking up a coalition and forming a new one. With this assumption, the Cost vector of unassigned MR tasks can be modified such that if the domain contains the robots that have been used together in their last task assignment, a cost reduction can be achieved by the amount set by the threshold, $\alpha$. This can be implemented as a clearing rule.

Finally, the strategies maintaining a uniform distribution of tasks to robots and leaving one robot idle for future tasks can be implemented by looking up the number of tasks each robot has in their agenda, before an assignment, and incurring penalties that violate the strategy.

In the case of leaving $k$ robots idle for future tasks, a penalty can be applied to assign-
ments that reduce the number of robots with 0 tasks assigned, to be $< k$ and this penalty can be proportional to the number of robots that violate this strategy.

A similar approach can be taken for uniformly distributing tasks to robots, by applying a penalty proportional to the sum of absolute differences between the number of agenda items of the robots.

The computational complexity of the proposed SSI method and integration of the above heuristics involves four steps:

1. computation of $Acc$ and $Cost$ vectors for each task

2. computation of $Dom$ vector for each task

3. computation of edge weights, $w_{ij}$ between tasks $t_i$ and $t_j$, for every $i$ and $j$.

4. sorting the tasks according to a heuristic function

One of the key computational problems when constructing a TAG is item (1), since accessibility and cost calculations of each pair of platform and task requires $O(nm)$ computations of running a path-finding algorithm for $n$ robots and $m$ tasks. Then, the bids for MR tasks need to be used to further build the domains of these tasks. This requires computing the combination of all the robots that have access to a task, $t_i$, therefore, $O(|Acc_i|)$, repeated $m$ times, for all the tasks in the TAG. Sorting the tasks according to $est$ or some other weighted heuristic function, as proposed, would require $O(m\log(m))$. Once sorted, the tasks can be allocated in order one at a time. In the proposed scheme using auctions, the calculation of the cost and accessibility can be delegated to the robot platforms, and these values can be populated on the side of the auctioneer via submitted bids. Furthermore, domain reduction techniques can be employed by selective bidding rules, where the robots do not send bids to all the tasks. Such situations might be beneficial when a problem is scaled and there are sufficient platforms to employ the strategy of partitioning the geographic area in responsibility regions.
The implementation and testing of this proposed method is left for future work.

5.4 Summary

In this chapter, I’ve covered the analysis of mission-level data produced during the TASC experiments. This data is divided into two parts, first, the data collected using the software, and second, the post-experiment interview sessions, which gave us insight into and explanations regarding participants’ thought processes during the supervision of the team.

For the first part, I’ve looked at objective metrics, such as run-time, for $TE$, interaction, activity and total deliberation times, as well as subjective metrics, namely the NASA-TLX score and its individual components. Then these metrics were compared between two groups of participants, divided based on the interaction modes that they used. In these analyses, it was found that there wasn’t any significant difference between the groups, for both objective and subjective performances. My conjecture is that the scenarios were not hard enough to observe a significant difference, as supported by the ratings provided by the NASA-TLX’s mental demand component. On the other hand, the group using collaborative allocation mode had significantly lower interaction, activity and total deliberation times, which means that the percent busy time of the operators was less than in manual allocation mode. As stated in Chapter 2.3, the percent busy time is a valid proxy for operator mental workload, which generally leads to higher performance. Therefore, I speculate that, through lower mental workload and more free time, the collaborative allocation mode may lead to better performance when the problem is scaled.

The second part of the analysis focused on the discussions and answers provided during the post-experiment interviews. During these interviews, participants were asked to comment on their assignment strategies. These strategies are listed and explained in Chapter 5.2.2, together with the responses provided in TCAT experiment. These strategies and the mission-level data provides the answer to RQ5, for the problem domains investigated in this dissertation.
The interaction between the operator and the decision support functionality was also examined both in subjective ratings provided at the end of each scenario and also during the post-experiment interview sessions. One important observation made during this analysis suggests that the trust in the autonomous capabilities of the planner and the ways that this tool have been utilized varies significantly, therefore is highly dependent on the personal expectations and preferences of the operators. This analysis also showed that one group of participants had strong negative experiences using the decision support functionality, while another group trusted the tool to the point of using it without fully understanding the plans produced in certain situations. This result leads to the obvious conclusion that in order to improve the mental workload and team performance, research effort has to be made to customize and steer the behavior of automated planners, which constitutes the key motive behind the approach adopted in this dissertation.

Finally, the integration of the strategies employed by the participants in TCAT and TASC and the TAG and MAP models to common MRTA methods were discussed. Specifically, the steering of CP techniques and auction mechanisms, using the strategies as heuristics, were examined, which describes a context in which RQ6 can be answered. Implementation of an automated planner using the SSI auction and the heuristics, as described in Chapter 5.3, and providing a full answer to RQ6 is left for future work.

In the next and final Chapter 6, I’ll review the contributions of this work, recapitulate the thesis statement, discuss how the research questions posed in this dissertation are addressed and mention the planned future work for the direction of this research.
Chapter 6

Conclusion

In this final chapter, I’ll first revisit the research questions and the thesis statement in Chapter 6.1, as posed in Chapter 1.2, and how they are answered in the chapters that followed. Next, in Chapter 6.2, the contributions of the dissertation will be stated. Then, in Chapter 6.3, the limitations of the models and methodology will be discussed. Finally, in Chapter 6.4, the planned and possible future directions for the work described in this dissertation will be described.

6.1 Revisiting Research Statement and Questions

In Chapter 1.2, I phrased the thesis statement as: Human-centric approaches for assisting the operator in human multi-robot teams for task and resource allocation can mitigate some circumstances in which performance would otherwise degrade due to declining situational awareness. This abstract form of the statement is clarified and more specific use of the terms “human-centric approach” and “assistance to the operator” for task and resource allocation are defined in Chapter 2.4. According to these definitions, human-centric approach refers to the modeling of the mental workload and decision making behavior of the human operators, and the assistance to the operator refers to decision support functionality and is discussed in the context of using automated planners to help operators solve task assignment and scheduling problems. Furthermore, the exact properties of the
task assignment problem are defined as complex task domains, referring to the existence of interdependencies among tasks and tight coordination requirements among robots, and are compared to the Generalized Assignment Problem (GAP) and its extended version, E-GAP.

The thesis statement is broken down into specific research questions and addressed in chapters 3-5, dividing the verification of the thesis statement into incremental steps. The first two research questions:

**RQ1** Can an intelligent HRI system be of measurable help to a human operator, when making decisions regarding allocation of tasks with constraints and dependencies?

**RQ2** How can an intelligent HRI system be guided according to operator expectations when reasoning about task/resource allocations in real-time, during high-stress missions?

are the core questions that define the approach adopted in this dissertation. RQ1 was answered in Chapter 5, through mission-level analysis of the TASC experiment. As these analyses revealed, even with the sub-optimal greedy allocation assignment algorithm, mental workload and interaction time were reduced significantly with the TASC interface compared to the control group who performed assignments manually. These results confirmed earlier findings of related work, for the problem domain and the system used in this dissertation. The real motivation behind this work lies in RQ2. In order to improve the interaction between the operator and the automated planners, a formal model is constructed to represent the human operators’ mental workload. Then this model and the strategies employed during the user studies are discussed in terms of how they can be used to guide the algorithms behind the autonomous planners. The model is formulated in two different environments, as specified by the research questions listed below:

**RQ3** What are the factors and to what degree do they affect a human supervisor’s decision making regarding task assignment under *stress-free and favorable circumstances*?

**RQ4** What are the factors and to what degree do they affect a human supervisor’s decision making
In order to address these research questions, I have developed two software tools, TCAT, described in Chapter 3.4, and TASC, described in Chapter 4.2, which were employed in the user studies. RQ3 is addressed in Chapter 3. In this chapter, the task landscape and the properties of tasks according to the work in this dissertation are described. The TAG model is defined and verified in the TCAT user study. Two of the factors, platform requirement and constraint properties of tasks, are represented via two metrics $APR$ and $CTR$, and were shown to affect the mental workload and perceived performance of the human operators. Hence RQ3 is answered for these two factors. RQ4 is addressed in Chapter 4. In this chapter, dynamic environments are considered. The TAG model, which describes an instance of a task assignment problem, is extended to describe multiple instances of task assignment problems and is defined as the MAP model. A MAP contains a sequence of TAGs representing decision points during a mission and indicate where a replanning took place. To verify the MAP model, the TASC experiment was carried out, and three factors were identified, namely the constraint property of the task ($CTR$), spatial distribution of the platforms ($ADD$) and the effects of plan disruption ($TDR$), and were shown to affect the operator mental workload. Therefore RQ4 has been answered for these three factors.

The TASC experiment also compared the mental workload and the performance of two groups of operators: one group explicitly assigned tasks manually and the other group had the option to delegate the task assignment to an automated decision support tool at any point during the mission. The goal of this comparison was to address the following research questions:

**RQ5** What are the common strategies adopted by the supervisors?

**RQ6** How can useful human strategies be integrated with common MRTA techniques?

These questions are addressed in Chapter 5. The data for the strategies commonly employed by participants were collected during post-experiment interview sessions. These interviews were conducted both for TCAT and TASC experiments, therefore the strategy and behavior shifts between
the static and dynamic domains are also observed. The overview of the strategies employed for
task allocation is presented in Chapter 5.2.2, answering RQ5. As described in that chapter, al-
though there were a lot of commonalities among the users’ approaches to the task assignment
problems, the conditions that these strategies were used differed significantly among the partici-
pants. These results strengthen the approach I have taken in this work to customize automated
planners. Chapter 5.3 discusses how these strategies can be used as heuristics to guide two common
MRTA methodologies used by the automated planners, namely auction mechanisms and constraint
programming. This discussion provides a starting point in order to fully answer RQ6. Answering
RQ3, RQ4, RQ5 and RQ6 collectively also answers RQ2, for the interaction scheme described under
the circumstances presented in the scenarios.

6.2 Contributions

In this section, I’ll recapitulate the contributions of the work presented in this dissertation, which
were first discussed in Chapter 1.4. As stated in that chapter, the main contribution of this work
is to the field of HRI, specifically in the context of multi-robot team supervision.

One of the contributions of this work is a human-centric taxonomy of task characteristics
and landscape, presented in Chapter 3.2. This taxonomy is based on the analysis of the existing
task properties commonly occurring in MRS literature and task taxonomies, as presented in Chap-
ter 3.1 and Appendix A. These existing taxonomies consider only task assignment problems from
a computational perspective. Since task allocation algorithms can handle a variety of constraint
types, from the perspective of these taxonomies, the task assignment problems are classified based
on the existence of constraints, without requiring further distinction of the types of these con-
straints. In other words, if the tasks are constrained, the solution requires modeling the problem
in a specific way and utilizing an algorithm capable of solving all problems with similar properties.
Once this is done, the type of the constraint, whether it is an ordering constraint or a specific
deadline, does not have a significant effect on the computation of the solution. However, from the human perspective, the task assignment problem is cognitively more complex, since humans tend not to strictly follow a particular algorithm for every scenario. This is also true even for a single scenario, where an operator sometimes utilizes multiple heuristics to reduce the solution space. The different types of constraints are not perceived in the same manner, therefore require different ways for visualization and interaction as discussed in Chapter 3.2. In my work, the constraints imposed on the tasks are further broken down into sub-categories based on how they originate, cognitive accessibility and possible interface requirements. I believe this to be a valuable contribution since such classifications can be used as guidelines to compare different systems in terms of interaction design and performance evaluation.

At the heart of this research lies the graph-based model TAG and the MAP, which is a sequence consisting of TAGs representing the decision points that cause replanning during a mission. I have developed two software applications as testbeds to empirically verify the metrics, which are derived from these models and each representing one of the investigated factors. First with the TCAT, described in Chapter 3.4, a user study was conducted and the TAG model’s two metrics, $APR$ and $CTR$, were verified. Then, with the TASC, described in Chapter 4.2, another user study was conducted and the MAP model’s three metrics were verified. The results from these experiments showed that all of the investigated factors have a significant effect on mental workload and the metrics representing these factors can be used to predict mental workload of human operators. The methodology to verify these models was empirical, a common method to verify methods in HRI since the cognitive models for human decision-making in this context are still not fully understood. I believe these models to be significant contributions to HRI, since they focus only on the spatial relationships between tasks and platforms that are capable of executing them. These properties are common to all types of MRS applications therefore, in this regard, these models are domain independent and they can be beneficial to SOMR control system designers in
any indoor problem environment utilizing UGVs. They may also extend to outdoor environments and UAVs, as well as mixed UGV/UAV teams, but testing these extensions was beyond the scope of this thesis.

Also, during these experiments, human strategies employed regarding decision-making in task allocation problems were observed both in time critical dynamic environments and in static domains. This allowed for comparison between strategies used in “ideal” situations, where planning time was not restricted, and “realistic” settings, where the robots had to execute their assigned tasks while the environment changed frequently. The results showed that the choice of strategies employed in these experiments varied significantly among participants (Chapter 5.2.2). For real-life missions in dynamic environments, forming and maintaining a complete plan at all times may not be expected from the operators, as was required in the dynamic environment (TASC) experiment (e.g., operators can assign important tasks first and figure out future steps later). However, this approach revealed the high mental workload involved during planning, even for thinking a few steps ahead. This result, together with the variance in strategies employed by participants, confirmed the importance of utilizing automated planners that can be examined and adjusted by human operators, prior to the actual missions, according to their own preferences and expectations. To this end, the implementation of the strategies developed by participants as heuristics to steer the behavior of two commonly employed classes of MRTA algorithms (constraint programming and auction mechanisms) was discussed in (Chapter 5.3). In addition, in the same Chapter, I explained how these heuristics can be integrated into automated planners using the TAG model and an SSI auction mechanism. I believe this to be another significant contribution of this dissertation, to improve the interaction between operators and automated planners for several reasons. First, the heuristics are domain independent since they only consider generalized task properties and spatial relationships, common to MRS application domains. Second, these heuristics are not designed but observed during user studies. Last, the suggested automated planner implementation is straight
forward and can be a guideline to build steerable automated decision support systems.

As stated in Chapter 1.4, the data produced during both experiments is valuable since user studies in multi-robot teamwork are hard to acquire, and the results are generalizable due to the detailed specification of the environmental and task characteristics based on the aforementioned task taxonomy and complexity studies. The collected data contains all interaction events, decisions, messages between the components and for the positions of robots for the entire duration of the experiment. Therefore the scenarios can be replayed and new metrics and analysis can be performed.

These contributions however strictly apply to situations where the assumptions of this work holds, which are discussed in the next section.

6.3 Limitations of the Methodology and the Models

The answers provided to the research questions addressed and the contributions of this dissertation are only valid under certain conditions that stem from the limitations of the methodology and the models. In this section, I’ll briefly describe the key points which require further research or need to be taken into account when evaluating the conclusions derived from the analysis of the studies.

In Chapter 3.2, I described the properties of the types of tasks with which this dissertation is concerned. Among the four properties described in Chapter 3.2, the requirements and constraint properties are studied, while the granularity and frequency properties are not represented or accounted for in either the TAG or MAP models.

The experiments were conducted for scenario maps for indoor environments. The reason for this was to reduce the cognitive complexity that could arise from unstructured maps, therefore participants not being able to fully grasp the map topology. This assumption was necessary to restrict unsystematic variance within the experiments so that the TAG and MAP models could be verified. In theory, there is no reason for the TAG and MAP models not to work in unstructured
environments as well, but these models were only verified in structured ones therefore, the claims made in this dissertation have only been proven in these environments.

Both the TCAT and TASC experiments found that $CTR$ has a significant effect on users’ perceived performance and mental workload, but only by a small margin. Yet the properties that have been used for labeling a task as critical, blocking access to tasks or resources, were repeatedly mentioned by users either as a factor for mental demand or used commonly to prioritize tasks in most of the participant-devised strategies. According to participants’ statements, the priority given to critical tasks and also constituent factors that make a task be regarded as critical by users varied significantly and changed during a mission. Due to these reasons, the exact factors that make a task viewed as critical depends on personal preferences and remains largely a mystery.

Construction of TAGs require computationally costly path-finding algorithms, not just from robots to tasks but also between tasks as well. This high computation cost is likely to be problematic for centralized systems for large-scale applications. As suggested in Chapter 5.3, however, it is possible to delegate this cost computation to robots in distributed settings.

One important fact to note when making inferences from the data collected is that the participant population may not represent the target population of some domains that these models are mainly designed for, such as USAR. The models were designed to be generic and can be applied to any domain that would require supervision of a team in environments bearing the spatial and task-based characteristics investigated here. Therefore participants were accepted from all backgrounds and ages and no restrictions or selection criteria were imposed. If these models were to specifically target a particular domain such as USAR and participants were to be chosen among target population (e.g. first responders), the results may be different.

Also, in order to be able to conduct experiments in reasonable time frames, certain decisions were made during experiment design which likely effected the data produced. The training sessions had to be kept short, therefore the interface followed a minimalist design, limiting the number of
possible ways to perform actions. One such example is assigning keyboard shortcuts to robots, which could speed up the assignment process but was not implemented. Also in order to test a high number of independent variables, the scenarios were kept short with a low number of robots and tasks. This made the scenarios unrealistically simple. Finally, among the factors and task properties that were investigated, the values were discretized into low and high values, therefore a continuous range of values were not analyzed.

Lastly, some of the parameters such as \( CTR \) are dependent on the situation and stress factors involved in critical real-life missions (e.g., USAR). Such conditions cannot be fully replicated in a lab environment. Therefore, the data produced in real mission environments would probably be different even if the relationships between the factors stay the same.

6.4 Future work

The possible future research based on the work described in this dissertation can be categorized into three directions: improvement of the models, investigation of decision support functionality and investigation of the problem domain.

6.4.1 The Models

The models can be improved in several ways. First, as stated in the previous section, the task properties, granularity and frequency, were specified but not included in the scope of this dissertation. Based on the possible applications of MRS teams, these properties are important, therefore extension of the TAG and MAP models could be beneficial in terms of extending their utilization to a larger domain of problems. In its current form, the TAG model is defined such that the cardinality of the sets in \( Dom_i \) is constant, which is represented as the \( req_i \). In order to account for the granularity property, a more flexible assignment model is needed, for which either tasks will have to be further decomposed or \( Dom \) will have to be changed into a more complex data structure to
accommodate a dynamic range of assignments with an associated utility model. For the frequency property, the MAP can be arranged in a way to auto generate new TAGs that include task vertices for the tasks that are required to repeat in certain intervals. This effectively changes the problem, for the repeated tasks, to time extended assignment (TA) instead of instantaneous assignment (IA), according to Gerkey and Mataric’s taxonomy [32].

Also, the model currently is designed for only implicit type constraints, described in the task properties in Chapter 3.2. In real-life missions, it is expected that the operators or higher authorities have the capability to impose constraints such as hard deadlines (temporal) or priority ordering (explicit) on the execution of tasks.

In order to feasibly conduct experiments and produce sufficient data to verify my models, only one metric per factor was tested, and only for high and low values. Existing metrics can be further tested over a continuous range of values to explore the relationships between the gradual increase of the metrics and the mental workload and perceived performance. This requires scaling up the problem and longer test sessions per metric.

Another addition to model improvement is through exploration of other metrics that can represent the factors verified to affect mental workload. Factors such as spatial distribution of platforms can be expressed using metrics other than ADD, or tasks can be labeled critical according to factors other than the ones described in Chapter 3.3. Devising additional metrics may yield better representation of these factors or complementary ones in certain domains.

### 6.4.2 Decision Support

As one of the main goals of the work described in this dissertation, improving the interaction between human operators and decision support systems, in the context of task allocation, requires additional research. So far, a model for operator decision making has been defined and described and a method to incorporate the metrics and the findings of the experiments conducted here has been
discussed in Chapter 5.3. One of the high priority, planned future activities is the implementation and empirical evaluation of this weight-based, human-guided decision support functionality.

Furthermore analysis for integration of the proposed method to other MRTA methods and algorithms would potentially provide additional areas for utilization, especially in systems in which such algorithms are already being used. One such method is Mixed-Integer Programming, used commonly in optimization problems and in some MAS and MRS research [73]. This investigation is beyond the scope of this dissertation since the methodology requires centralized solvers.

As covered in Chapter 2.3, there is related research in which decision support systems have been utilized in the same context as discussed in this dissertation. As covered briefly, the systems used in the research described employ a variety of methods to establish good interaction between the human operator, such as extra interface components to form and display “what if” scenarios, before approving a plan of execution. Similarly, further research in designing interaction schemes to steer decision support systems is required.

6.4.3 Problem Domain

The models described and verified in this dissertation was only used in a toy-problem domain, designed similarly to the RoboCup Rescue Simulation [50], as described in Chapter 3 and Chapter 4. Testing the models in different environments warrants further research. The main changes to the environment for further investigation that I see as a priority are changing the problem definition, scaling it up to include more robots and tasks, and other dynamic elements of the domain such as task arrival rates, changes in the spatial map and platform capability and availability.

By changing the problem definition to something other than the USAR domain, the variance in the results for mental workload and perceived performance can highlight the effect of scenario context on operators’ decisions. For example, blocking tasks may not be seen as high priority tasks in a different domain such as stock organization in a warehouse, since in such environments,
additional task locations are predesignated.

As the problem scales in size, in terms of more robots and tasks, the mental workload of the operator is expected to increase. It would be interesting to investigate operators’ behavior in situations where no solution exists that match all mission parameters or constraints. Also the strategies employed and the interaction with decision support functionality may change significantly in large scale problems.

Finally the dynamic properties vary for all environments. In the TASC experiment in Chapter 4, the scenarios were set to introduce new random tasks in fixed time intervals (45 seconds). It is easy to change this frequency to overwhelm the operator by either increasing the number of tasks added to the problem domain or change the frequency of change or even removing the predictability of new task arrival by making it random. Further changes to the environment and the problem domain may include changes in the map properties, such as addition or removal of obstacles, changes in the nature of the tasks (e.g., fire extinguishing tasks spreading over the area when left unattended) or changes in the platform availability. All these properties change the problem in a significant way, therefore the behavior of the operators changes, as well as the algorithms and techniques used for decision support functionalities. For the models described in this dissertation, these changes require further investigation.

6.5 Summary

In this dissertation, I’ve described two models, the TAG and MAP models, which are used to analyze and predict the complexity of task assignment problems, from the human operator’s perspective, in static and dynamic domains respectively. Both models were empirically verified through user studies conducted using the HRTeam framework and two software tools, TCAT and TASC, which were developed for these experiments. The TAG model examined two factors represented by $APR$ and $CTR$ metrics. Both these metrics were proven to affect the mental workload and perceived
performance of the human operators. The MAP model extended the TAG model to encompass factors in dynamic environments and examined three factors represented by CTR, ADD and TDR metrics. All of the metrics were proven to affect both the objective mental workload, measured as plan construction time, and subjective mental workload of the human operators. Because these models were built based on domain independent characteristics, such as spatial distribution of resources and tasks, they can be applied to a variety of SOMR and MRS applications.

Aside from the verification of the TAG and MAP models, the user studies also focused on and analyzing the interaction between the operators and the autonomous planners. For the latter purpose, the TASC interface was designed to allow for two different levels of autonomy for the interaction scheme: manual control and collaborative control, with an option to invoke an automated assignment tool. Most participants found the automated assignment functionality useful. Although the auto-assign used a suboptimal greedy algorithm, the performance of the collaborative allocation group are found to be similar to the manual allocation group’s performance, with reduced mental workload and interaction time with the interface.

Strategies employed by the participants during these experiments were analyzed in detail and discussed in terms of how they can be used as heuristics to steer two commonly employed classes of MRTA algorithms. Finally, how these heuristics can be integrated into automated planners using the TAG model and an SSI auction mechanism, is explained.
Appendix A

Task Analysis

This section contains a classification of tasks according to the description in Section 3.2, which appear in literature. The list is not exhaustive.

**Abbreviations:**

- **R** (required platforms): single-robot (SR) or multi-robot (MR)
- **C** (constraints): independent (ID) or constrained (CN)
- **F** (frequency): achievement (AC) or maintenance (MN)
- **G** (granularity): atomic (AT) or composite (CP)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
<th>Setup</th>
<th>R</th>
<th>C</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bejar et al.</td>
<td>SensorCSP for wireless sensor tracking system.</td>
<td>N/A</td>
<td>MR</td>
<td>AC</td>
<td>CN</td>
<td>AT</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
<th>Setup</th>
<th>R</th>
<th>C</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dias and Stenz [22]</td>
<td>Distributed traveling salesman problem. Multi-round, single-item, first-price auctions by a central auctioneer and robots.</td>
<td>Sim.</td>
<td>SR</td>
<td>AC</td>
<td>ID</td>
<td>AT</td>
</tr>
<tr>
<td>Frias-Martinez et al. [29]</td>
<td>RoboCup 4-legged soccer league. Role allocation via single-round, single-item, first-price auctions and combinatorial auctions. Central auctioneer.</td>
<td>Phy.</td>
<td>SR</td>
<td>MN</td>
<td>ID</td>
<td>CP</td>
</tr>
<tr>
<td>Gerkey and Matarić [31]</td>
<td>Object tracking (find and follow a colored object), sentry-duty (go to point and watch), clean up (locate and push a box to the edge of the room), and monitor object.</td>
<td>Phy.</td>
<td>SR</td>
<td>AC</td>
<td>ID</td>
<td>AT</td>
</tr>
<tr>
<td>Box pushing</td>
<td></td>
<td>MR</td>
<td>AC</td>
<td>ID</td>
<td>CP</td>
<td></td>
</tr>
<tr>
<td>Goodrich et al.</td>
<td>Guiding a robot through a maze.</td>
<td>Sim.: 1 robot</td>
<td>SR</td>
<td>AC</td>
<td>ID</td>
<td>AT</td>
</tr>
</tbody>
</table>

(Continued on Next Page...)

148
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
<th>Setup</th>
<th>R</th>
<th>C</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardin and Goodrich</td>
<td>Comparison of mixed-initiative, adjustable and adaptive autonomy modes in wide area search and rescue domain. Selection of regions to search, sweep region and item classification.</td>
<td>Sim.: 200 robots</td>
<td>MR</td>
<td>AC</td>
<td>ID</td>
<td>CP</td>
</tr>
<tr>
<td>Heger &amp; Heger et al. [42]</td>
<td>Software architecture and sliding autonomy investigation for joint performance of tightly coordinated tasks. Large structure assembly task. Required: docking beams to assemble a square structure.</td>
<td>Phy.: 3 robots</td>
<td>MR</td>
<td>AC</td>
<td>CN</td>
<td>CP</td>
</tr>
<tr>
<td>Reference</td>
<td>Description</td>
<td>Setup</td>
<td>R</td>
<td>C</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>Landén et al. [54]</td>
<td>UAV: Complex task allocation framework. Scenario: cooperative scan + delivering relief packages. Operator generates TST to define overall mission objectives</td>
<td>Phy.</td>
<td>SR</td>
<td>AC</td>
<td>CN</td>
<td>CP</td>
</tr>
<tr>
<td>Scerri et al. [77]</td>
<td>Multi-robot exploration, victim search. Weak cooperation: victims can only be reported on mapped areas and when UGV’s are in communication range.</td>
<td>N/A</td>
<td>SR</td>
<td>AC</td>
<td>CN</td>
<td>AT</td>
</tr>
<tr>
<td>Sklar et al. [89]</td>
<td>Multi-robot exploration. Sensor-sweep at goal point</td>
<td>Phy. + Sim.: Player/Stage</td>
<td>SR</td>
<td>A</td>
<td>ID</td>
<td>AT</td>
</tr>
<tr>
<td>Tovey et al. [95]</td>
<td>Multi-robot exploration. Sensor-sweep at goal point</td>
<td>Sim.: Player/Stage</td>
<td>SR</td>
<td>AC</td>
<td>ID</td>
<td>AT</td>
</tr>
<tr>
<td>Wang [96]</td>
<td>Victim search. Sensor-sweep at goal point &amp; teleoperation</td>
<td>Sim.: USARSim</td>
<td>SR</td>
<td>AC</td>
<td>ID</td>
<td>AT</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
<th>Setup</th>
<th>R</th>
<th>C</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Box pushing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mapping and victim search. Victims can only be marked on mapped areas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wang et al.</td>
<td>Multi-robot exploration. Sensor-sweep at goal point + teleoperation</td>
<td>Sim.: US-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[97]</td>
<td></td>
<td>ARSim</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[103]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

TCAT User Study Procedure

During the TCAT user study, for each participant the following procedure was followed:

1. Completion of the IRB consent form

2. Information about the experiments and instructions

3. Training Session

4. Q & A

5. Experiment

6. Post-experiment Interview

B.1 Instructions

Following the completion of the IRB consent form the following information was provided to each participant prior to the training session:

A Experiment:

- This session will be recorded.
The purpose of the experiment is to study human cognitive load when supervising a team of robots.

We’ll start with a training session followed by 27 scenarios

B Scenario

You will be presented with a floor plan of a building after a disaster.

You will come up with a plan for a team of robots, where you will specify which robot is going to do which of tasks and the order they will be carried out.

Tasks may either be:

* fire where robots need to extinguish,
* debris caused by the structural damage that needs to be removed or
* sensor sweep where robots need to go to a point and collect information

All tasks are of equal priority.

Some tasks will require more than a single robot.

Once all of the robots assigned to a task reach it, the task execution will take a fixed 1 second regardless of the type or the number of robots required for the completion of the task.

C Goal

Your goal is to come up with an execution plan that would complete all of the tasks as fast as possible, if it were to be executed.

D Surveys

At the end of every scenario we will ask you to tell us how hard was it to come up with the plan and how confident that your plan will be effective.
– After you complete all of the experiments we will finish with a debriefing and interview session.

B.2 Post-Experiment Interview Questions

The following questions were asked to each participant, upon completion of the study:

1. How was your experience with the tool? Was it easy/hard to work with? Any comments about your experience?

2. After going through all the different scenario types, which one’s do you think stand out as harder ones?

3. Did you had a particular strategy for allocating the tasks? For example when you encountered a scenario for the first time did you have a fixed pattern or started the same way? Or did you had several for different situations?
Appendix C

TCAT Scenarios

The scenarios used in TCAT experiments are shown in this section. The presentation order follows the category labeling Q1-Q4 as described in section 3.

C.1 Category Q1 - CTR low, APR low

Figure C.1: TCAT Scenario-1
Figure C.2: TCAT Scenario-2

Figure C.3: TCAT Scenario-3
Figure C.4: TCAT Scenario-5

Figure C.5: TCAT Scenario-14
C.2 Category Q2 - CTR high, APR low

Figure C.6: TCAT Scenario-15

Figure C.7: TCAT Scenario-6
Figure C.8: TCAT Scenario-7

Figure C.9: TCAT Scenario-11
Figure C.10: TCAT Scenario-16

Figure C.11: TCAT Scenario-18
C.3 Category Q3 - CTR low, APR high
Figure C.14: TCAT Scenario-20

Figure C.15: TCAT Scenario-21
Figure C.16: TCAT Scenario-24

Figure C.17: TCAT Scenario-26
C.4 Category Q4 - CTR high, APR high
Figure C.22: TCAT Scenario-12

Figure C.23: TCAT Scenario-13
Figure C.24: TCAT Scenario-17

Figure C.25: TCAT Scenario-19
Figure C.26: TCAT Scenario-22

Figure C.27: TCAT Scenario-23
Appendix D

TASC User Study Procedure

During the TASC user study, for each participant the following procedure was followed:

1. Completion of the IRB consent form

2. Information about the experiments and instructions

3. Training Session

4. Q & A

5. Experiment

6. Post-experiment Interview

D.1 Instructions

Following the completion of the IRB consent form the following information was provided to each participant prior to the training session:

A Experiment

- This session will be recorded.
The purpose of the experiment is to study human cognitive load when supervising a team of robots.

We’ll start with a training session followed by 8 scenarios

B Scenario

– You will be presented with a floor plan of a building after a disaster.
– You will come up with a plan for a team of robots, where you will specify which robot is going to do which of tasks and the order they will be carried out.
– Tasks may either be:
  * fire where robots need to extinguish,
  * debris caused by the structural damage that needs to be removed or
  * sensor sweep where robots need to go to a point and collect information
– All tasks are of equal priority.
– Some tasks will require more than a single robot.
– Some of the tasks will be available as soon as the scenario starts and others will become available later in fixed time intervals.
– Once all of the robots assigned to a task reach it, the task execution will take a fixed 10 seconds regardless of the type or the number of robots required for the completion of the task.

C Goal

– Your goal is to supervise robots to finish all of the tasks as fast as possible by assigning them to tasks or in other words coming up with the plan of execution.
– In this experiment, your job is to ensure that a plan composed of a full assignment of robots to tasks exists at all times.
Although it may be tempting to assign some of the tasks and leave others for later (since the situation may change drastically over time) this strategy will prevent us from observing the decision making behavior that we seek to observe in this experiment. Therefore it is prohibited.

D Interaction Mode

- You will assign robots to tasks, manually in 4 of the 8 scenarios and collaboratively with an automated assignment functionality, in the remaining 4 scenarios.
- In collaborative mode, the automated assignment functionality may not produce the most effective plans. It is your responsibility to correct the plans that the coordinator produces.
- The scenarios will appear in alternating interaction modes, one in manual then in collaborative mode.

E Surveys

- At the end of every scenario we will ask you to tell us how hard was it to come up with the plan and how confident that your plan will be effective. If the interaction mode was collaborative an additional short survey will be presented regarding your thoughts on the automated assignment functionality.
- After you complete all of the experiments we will finish with a debriefing and interview session.

D.2 Post-Experiment Interview Questions

The following questions were asked to each participant, upon completion of the study:
1. How was your experience with the tool? Was it easy/hard to work with? Any comments about your experience?

2. After going through all the different scenario types, which one’s do you think stand out as harder ones?

3. Did you had a particular strategy for allocating the tasks? For example when you encountered a scenario for the first time did you have a fixed pattern or started the same way? Or did you had several for different situations?

4. In collaborative control mode, what were the particular situations when you felt the need for support therefore left the decision to the coordinator agent. Did you had a particular strategy when utilizing the coordinator agent.
Appendix E

TASC Interface

In this section an example scenario walkthrough in TASC interface is presented. TASC interface starts with a screen providing general instructions regarding the experiment and collects participant information, specifically user's designated ID for the experiment, age, gender and computer experience in years (Figure E.1).

Figure E.1: TASC Initial screen
The initial screen is followed by further instructions regarding the scenarios in the experiment as seen in Figures E.2 and E.3. This screen also notifies user about the scenario type (training or experiment) and its number. The mode of operation is also notified either as MANUAL or AUTO COMPLETE.

Figure E.2: TASC Training instruction screen.

Figure E.3: TASC Experiment instruction screen.

In order to make sure that the users are aware of the mode of operation, a pop-up window
is presented to the users asking the mode of operation. Upon providing an answer, regardless of
the choice, correct answer is displayed before the start of the scenario (Figure E.4).

![TASC Interaction Mode Check](image)

Figure E.4: **TASC Popup reminder for the interaction mode**

A sample scenario beginning is shown in Figure E.5. The interface has a map component
displayed on the upper-left portion of the window, which shows the task and robot positions as
well as the paths that will be followed by the robots. In Figure E.5, robot *b-12* is assigned and on
its way to task *t-1* (a debris removal task). The immediate paths of robots are displayed in bold
green color.

![TASC Example: Scenario beginning. Robot b-12 is assigned to task T-1 and moving towards it.](image)

Figure E.5: **TASC Example: Scenario beginning. Robot b-12 is assigned to task T-1 and moving towards it.**

The allocation component is displayed on the upper-right portion of the window, where the
users can add/remove/modify assignments. In Figure E.5, the robot $b-12$ is shown to be linked with the task $T-1$, shown as a square. The numbers in the task square $x/y$, mean that this task still needs $x$ robots out of $y$ robots required to complete the task. In this figure $T-1$ still needs 1 more robot to complete and the task requires 2 robots overall.

The bottom-left portion shows the expected completion time of the tasks. In Figure E.5 this area is empty since none of the tasks are yet fully assigned. The bottom-right portion displays a clock and wait/resume, clear and auto assign buttons.

Figure E.6: All tasks are fully assigned. The paths that will be followed by the robots are displayed in bold green. The paths that are shown in gray represent the path to the next task of the robots.

Figure E.6 shows a full assignment, where robots $b-12$ and $b-13$ are assigned to task $T-1$ and robot $b-12$ is assigned to task $T-2$. The timeline component on the bottom-left shows the expected completion time of the tasks. In this example $b-12$ is expected to reach task $T-1$ first (shown by the short solid line ending with an arrow) then wait for $b-13$ to arrive (shown as the dotted line) to start executing the task. Then $b-12$ will move towards $T-2$. Both tasks are expected to complete in 1 min 34 seconds, displayed on the top-right portion of the timeline component.
Fully assigned tasks are displayed dimmed in both map and allocation components.

Figure E.7 shows the state of the mission after the arrival of a sensor-sweep tasks (T-3) and a multi-robot fire extinguish task (T-4). This figure also shows that the robot b-12 has arrived at T-1 and waiting for b-13. The clock icon displayed on top left corner of b-12 represents waiting state of the robots.

Figure E.7: *TASC Example: State of the mission after tasks T-3 and T-4 arrive.*

In figure E.8, robots b-12 and b-13 arrived at their assigned task T-1 and start executing it, signified by the triangle shape appears on top-left corner of the robot icons. Similarly b-14 is almost at its destination. The gray path lines show the trajectories of the robots b-12 and b-13 upon completion of task T-1.
Figure E.8: Robots b-12 and b-13 began execution of T-1. b-14 is just about to arrive T-4.

Figure E.9 shows the status after completion of T-1 and T-4, new tasks T-5 (a debris removal task that blocks the corridor area) and T-6 (sensor-sweep task) arrive.

During planning, when there are overlapping trajectories, users often want to see a particular robot’s assigned tasks and planned paths to reach its targets. This is done by hoovering over the robot icon with the mouse pointer in the allocation component, which in turn highlights the planned paths for that robot in the map component with a bright green color as shown in figure E.10.
Figure E.9: TASC Example: New tasks T-5 and T-6 arrive.

Figure E.10: TASC Example: Highlighting a robot's planned trajectories.
The "Wait" button enables users to stop the robots from following their paths. Once pressed the button label changes to "Resume", as can be seen in figure E.11. Pressing the "Resume" button allows the robots to start moving again towards their destinations. This button was intended as a "panic button" to allow users to modify the plan while robots remained in a fixed location if they felt overwhelmed. However, this button did not pause the experiment or stopped the clock, thus it was rarely used by the participants during TASC experiment. Also shown in figure E.11, last two tasks in this scenario, T-7 a sensor-sweep task and T-8 a 2 robot fire extinguishing task arrive.
In the remainder of this scenario, first the tasks \( T-7 \) and \( T-8 \) are assigned as shown in figure E.12. This is done by using the "Auto Assign" button, which only considers assigning tasks that are not completed by the user. Therefore in this situation, loyal to the existing plan, autonomous planner completes the assignment in a sub-optimal manner.

Once the task \( T-5 \) is completed (figure E.13), the plan is revised by first pressing "Clear" button and then "Auto Assign". Because "Clear" button removes all assignments the "Auto Assign" has no obligation to be loyal to any assignments made by the user, therefore computes a better plan.

Finally, in figure E.14 robots approach to their last tasks to complete the mission.
Upon completion of each scenario, users are presented with the NasaTlx [1] survey as seen in figures E.15, E.16 and E.17. The figures show our implementation of the NasaTlx survey into TASC interface in order to expedite the experiments, using exactly the same descriptions and procedure as the original version.
Figure E.15: TASC Example: NasaTlx Instructions

Figure E.16: TASC Example: NasaTlx
If the users have operated in relinquish control mode, an additional survey window is presented to users for subjective evaluation of the auto complete tool, regarding its usefulness and performance (figure E.18).
Appendix F

TASC Scenarios

This section contains the scenarios presented to participants of TASC user study. Each scenario contained 8 tasks in total and tasks were introduced 2 at a time in 45 seconds interval. A figure shows the situation of the scenario after introduction of new tasks. Scenarios are labeled from 5 to 12 (Scenarios 0-4 are for training).

F.1 Scenario-5

![Figure F.1: TASC Scenario-5 TAG1](image)
Figure F.2:  *TASC Scenario-5 TAG*$_2$

Figure F.3:  *TASC Scenario-5 TAG*$_3$
F.2 Scenario-6

Figure F.5: TASC Scenario-6 $TAG_1$
Figure F.6: TASC Scenario-6 TAG$_2$

Figure F.7: TASC Scenario-6 TAG$_3$
F.3 Scenario-7

Figure F.8: TASC Scenario-6 $TAG_4$

Figure F.9: TASC Scenario-7 $TAG_1$
Figure F.10: TASC Scenario-7 $TAG_2$

Figure F.11: TASC Scenario-7 $TAG_3$
F.4 Scenario-8

Figure F.12: TASC Scenario-7 TAG$_4$

Figure F.13: TASC Scenario-8 TAG$_1$
Figure F.14: *TASC Scenario-8 TAG*$_2$

Figure F.15: *TASC Scenario-8 TAG*$_3$
Figure F.16: *TASC Scenario-8 TAG$_4$*

**F.5 Scenario-9**

Figure F.17: *TASC Scenario-9 TAG$_1$*
Figure F.18: *TASC Scenario-9 TAG*$_2$

Figure F.19: *TASC Scenario-9 TAG*$_3$
F.6 Scenario-10

Figure F.20: TASC Scenario-9 $TAG_4$

Figure F.21: TASC Scenario-10 $TAG_1$
Figure F.22: TASC Scenario-10 TAG\textsubscript{2}

Figure F.23: TASC Scenario-10 TAG\textsubscript{3}
Figure F.24: TASC Scenario-10 TAG$_4$

F.7 Scenario-11

Figure F.25: TASC Scenario-11 TAG$_1$
Figure F.26: TASC Scenario-11 TAG$_2$

Figure F.27: TASC Scenario-11 TAG$_3$
Figure F.28: TASC Scenario-11 $TAG_4$

F.8 Scenario-12

Figure F.29: TASC Scenario-12 $TAG_1$
Figure F.30: TASC Scenario-12 TAG$_2$

Figure F.31: TASC Scenario-12 TAG$_3$
Figure F.32: **TASC Scenario-12 TAG$_4$**
Bibliography


