Intent Recognition in Multi-Agent Systems:

Collective Box Pushing and Cow Herding

by Najla Ahmad

University of Kansas, 2013

Submitted to the graduate degree program in Electrical Engineering & Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairperson: Dr. Arvin Agah

Dr. Victor Frost

Dr. Jerzy Grzymala-Busse

Dr. Sarah L. Kieweg

Dr. Bo Luo

Date Defended: December 13, 2013

The Dissertation Committee for Najla Ahmad

certifies that this is the approved version of the following dissertation:

Intent Recognition in Multi-Agent Systems:

Collective Box Pushing and Cow Herding

____

Chairperson: Dr. Arvin Agah

Date approved: __________________

Abstract

In a multi-agent system, an idle agent may be available to assist other agents in the system. An agent architecture called intent recognition is proposed to accomplish this with minimal communication. In order to assist other agents in the system, an agent performing recognition observes the tasks other agents are performing. Unlike the much studied field of plan recognition, the overall intent of an agent is recognized instead of a specific plan. The observing agent may use capabilities that it has not observed. This study focuses on the key research questions of: (1) What are intent recognition systems? (2) How can these be used in order to have agents autonomously assist each other effectively and efficiently? A conceptual framework is proposed for intent recognition systems. An implementation of the conceptual framework is tested and evaluated. We hypothesize that using intent recognition in a multi-agent system increases utility (where utility is domain specific) and decreases the amount of communication. We test our hypotheses using two experimental series in the domains of Box Pushing, where agents attempt to push boxes to specified locations; and Cow Herding, where agents attempt to herd cow agents into team corrals. A set of metrics, including task time and number of communications, is used to compare the performance of plan recognition and intent recognition. In both sets of experimental series, intent recognition agents communicate fewer times than plan recognition agents. In addition, unlike plan recognition, when agents use the novel approach of intent recognition, they select unobserved actions to perform, which was seen in both experimental series. Intent recognition agents were also able to outperform plan recognition agents by sometimes reducing task completion time in the Box Pushing domain and consistently scoring more points in the Cow Herding domain. This research shows that under certain conditions, an intent recognition system is more efficient than a plan recognition system. The advantage of intent recognition over plan recognition becomes more apparent in complex domains.

Key Words: Multi-Agent Systems, Multi-Agent Methodology, Multi-Agent Design, Plan Recognition, Intent Recognition, Distributed Systems, Collective Box Pushing, Cow Herding, Repast, Cooperative Multi-Agent Systems, Cooperative Agents

Acknowledgements

I would like to thank my advisor Dr. Arvin Agah. With just the right mix of encouragement, advice, and timely reminders, he was my guiding light whenever I felt like I couldn't find my way. I have been blessed by having fantastic advisors throughout my academic career, but I especially enjoyed putting my dissertation together under the watchful eyes of Dr. Agah.

Thank you to Dr. Victor Frost, Dr. Jerzy Grzymala-Busse, Dr. Sarah L. Kieweg, and Dr. Bo Luo for serving on my doctoral committee.

While there are numerous people who have shaped the course of my academic career, there are few in particular that I'd specifically like to mention.

It was my work with Dr. Costas Tsatsoulis, who guided me through a NASA supervised research position, which piqued my interest in multi-agent systems and encouraged me to explore intent recognition as a topic for my doctoral work. At that time, I also worked with Ed Komp, who was always there to give encouragement and technical advice.

My journey to become a computer scientist would never have happened without the support of Dr. Franklyn Turbak, Dr. Scott Anderson, and Dr. Allen Downey. These wonderful professors introduced me to the world of computer science, and showed me how it combined my love of math, science, problem solving, and art.

I would not be who I am today without my friends and family. I can't name all of them here, but there are many people to whom I owe a debt of gratitude.

I would like to especially thank my husband, Roni Talukdar, whose unwavering support has allowed me to achieve everything I have wanted so far. My mother, Fareena Sultan, was my inspiration to pursue higher education when I was a child and has helped me throughout my academic career. I would also like to thank my brother Beejul Khatri, who understands me better than anyone else, my father, Khalil Khatri, for his unending support and humor, and my grandparents, who played a huge role in raising me. I could not have completed the many years of my academic career without Hinna Mir and Gianni Evangelista, who blur the line between friends and family.

Table of Contents

Abstract iii
Acknowledgements v
List of Figures x
List of Tables xii
Chapter 1 Introduction 1
  1.1 Motivation 4
  1.2 Research Hypotheses 5
  1.3 Experiment Series 1: Box Pushing 5
  1.4 Experiment Series 2: Cow Herding 6
Chapter 2 Background and Related Work 7
  2.1 Multi-Agent Systems 8
  2.2 Plan Recognition 9
  2.3 Modeling Agents 15
  2.4 Plan Representation 20
  2.5 Time 24
  2.6 Communication Decisions 25
  2.7 Intent 32
  2.8 Other Approaches 35
  2.9 Multi-Agent Contest 35
    2.9.1 Food Gatherers 36
    2.9.2 Goldminers 36
    2.9.3 Agents on Mars 38
    2.9.4 Cows and Cowboys / Cow Herding 39
    2.9.5 Cow Behavior 41
    2.9.6 Cow Herding Strategies 42
  2.10 Related Work Summary 44
Chapter 3 Intent Recognition 45
  3.1 Intent Recognition Structure 48
Chapter 4 Research Methodology: Box Pushing 51

4.1 Experimental Setup 51
4.2 Agents 56
4.2.1. Types of Agents 57
4.2.2 Energy 58
4.3 Recognition 59
4.3.1 Plan Recognition 59
4.3.2 Intent Recognition 60
4.4 Actions, Plans and the Plan Library 62
4.4.1 Actions 62
4.4.2 Plans 62
4.4.3 Plan Library 63
4.5 Variables 64
4.5.1 Dependent Variables 65
4.5.2 Independent Variables 66
4.6 Complex Scenarios 67
Chapter 5 Experimental Result: Box Pushing 69
5.1 Communication 69
5.2 Time 70
5.3 Percent Completion 72
5.4 Accuracy of Recognition 73
Chapter 6 Research Methodology: Cow Herding 76
6.1 Agents 77
6.2 Cows 78
6.3 Complex Actions 78
6.4 Plan Library 79
6.5 Field 80
6.6 Team Organization 80
6.7 Time To Communication 81
6.8 Datasets 81
Chapter 7 Experimental Results: Cow Herding 83
7.1 Overview of Results 83

7.1.1 Total Wins 83
7.1.2 Overall Score 85
7.1.3 Differences Between Team Types 87
7.2 Recognition 88
7.2.1 Number of Recognition Agents Per Team 88
7.2.2 Plan followed by Observed Agents 89
7.2.3 Recognized Plans 92
7.2.4 Mean Recognition Time 97
7.3 Communication 97
7.3.1 Number of Communications 98
7.3.2 Time to Communication 98
7.4 Adding "Helper" Agents 100
Conclusion 102
8.1 Analysis of Hypotheses: Box Pushing 103
8.2 Analysis of Hypotheses: Cow Herding 104
8.3 Overall Results 106
8.4 Theoretical Contribution 106
8.5 Limitations 107
8.6 Future Work 108
Appendix 110
References 132

List of Figures

FIGURE 2.1: EXAMPLE IMAGE FROM 2008 COW HERDING SCENARIO DOCUMENT (DIX ET AL., 2008). 40

FIGURE 2.2: COW ALGORITHM 1 LOCATED IN 2010 MAPC SCENARIO DOCUMENT (DIX *ET AL.*, 2010). 41

FIGURE 2.3: COW ALGORITHM 2 LOCATED IN MAPC 2010 SCENARIO DOCUMENT (DIX *ET AL.*, 2010). 41

FIGURE 3.1: RECOGNITION STRUCTURE OVERVIEW. 45

FIGURE 3.2: THE PLAN RECOGNITION PROCESS. 46

FIGURE 3.3: THE INTENT RECOGNITION PROCESS. 47

FIGURE 4.1: SCREENSHOT OF THE BOX PUSHING SIMULATION. 52

FIGURE 5.1: COMMUNICATING AGENTS: IR (INTENT RECOGNITION), PR (PLAN RECOGNITION). 70

FIGURE 5.2: AVERAGE TASK COMPLETION TIME BY DATASET AND RECOGNITION TYPE: NR (NO RECOGNITION), PR (PLAN RECOGNITION), IR (INTENT RECOGNITION). RECOGNTION TYPE IS STATISTICALLY SIGNIFICANT IN DATASETS 1-9. 71

FIGURE 5.3: AVERAGE PERCENT COMPLETION BY DATASET AND RECOGNITION TYPE: NR (NO RECOGNITION), PR (PLAN RECOGNITION), IR (INTENT RECOGNITION). 72

FIGURE 6.1: SCREEN CAPTURE OF EXPERIMENT SERIES 2: COW HERDING. 80

FIGURE 7.1: NUMBER OF WINS FOR INTENT RECOGNITION TEAMS VERSUS PLAN RECOGNITION TEAMS. 84

FIGURE 7.2: THE NUMBER OF WINS FOR PLAN RECOGNITION TEAMS VERSUS TEAMS WITHOUT RECOGNITION. 84

FIGURE 7.3: NUMBER OF WINS FOR INTENT RECOGNITION TEAMS VERSUS TEAMS WITHOUT RECOGNITION. 85

FIGURE 7.4: TOTAL POINTS SCORED FOR INTENT RECOGNITION TEAMS VERSUS PLAN RECOGNITION TEAMS. 86

FIGURE 7.5: TOTAL POINTS SCORED FOR INTENT RECOGNITION TEAMS VERSUS TEAMS WITHOUT RECOGNITION. 86

FIGURE 7.6: TOTAL POINTS SCORED FOR PLAN RECOGNITION TEAMS VERSUS TEAMS WITHOUT RECOGNITION. 87

FIGURE 7.7: PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING PLAN 2 AND THE ALLOTTED TIME FOR RECOGNITION WAS 200 TIMESTEPS. 93

FIGURE 7.8: PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING PLAN 2 AND THE ALLOTTED TIME FOR RECOGNITION WAS 50 TIMESTEPS. 93

FIGURE 7.9: PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING PLAN 4 AND THE ALLOTTED TIME FOR RECOGNITION WAS 200 TIMESTEPS. 94

FIGURE 7.10: PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING PLAN 4 AND THE ALLOTTED TIME FOR RECOGNITION WAS 50 TIMESTEPS. 95

FIGURE 7.11: EXPERIMENT 2. THE PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING A RANDOM PLAN FROM THE PLAN LIBRARY AND THE ALLOTTED TIME FOR RECOGNITION WAS 200 TIMESTEPS. 96
FIGURE 7.12: EXPERIMENT 2. THE PLANS RECOGNIZED BY PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS IN THE CASE WHERE THE OBSERVED AGENTS WERE FOLLOWING A RANDOM PLAN FROM THE PLAN LIBRARY AND THE ALLOTTED TIME FOR RECOGNITION WAS 50 TIMESTEPS. 96
FIGURE 7.13: THE MEAN RECOGNITION TIME OF PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS BY DATASET. INTENT RECOGNITION HAS A STATISTICALLY LOWER MEAN RECOGNITION TIME IN ALL DATASETS. 97
FIGURE 7.14: THE NUMBER OF TIMES PLAN RECOGNITION AGENTS AND INTENT RECOGNITION AGENTS COMMUNICATED PER DATASET. 98
FIGURE 0.1: SCREEN CAPTURES FROM COW HERDING SIMULATION. SHOWS TIMESTEPS 0, 25, 50, AND 100. ALL AGENTS BEGIN IN A RANDOM LOCATION. 131

List of Tables

TABLE 3.1: EXAMPLE OF AN INTENT RECOGNITION AGENT'S PLAN LIBRARY. 49
TABLE 3.2: EXAMPLE OF OBSERVATIONS MADE BY AN INTENT RECOGNITION AGENT. 50
TABLE 4.1: EXPERIMENT SERIES 1 EXPERIMENTAL PARAMETERS. TOTAL EXPERIMENTS RUN: 3X6X5X11 = 990. 53
TABLE 4.2: EXPERIMENT SERIES 1 DATASET VALUES. 53
TABLE 4.3: BOX PUSHING: STARTING PLANS OF THE OBSERVED AGENTS. 55
TABLE 4.4: EXPERIMENT SERIES 1 BOX PUSHING - ACTION COSTS. 58
TABLE 4.5: EXAMPLE OF A PLAN LIBRARY FOR A BOX PUSHING PLAN RECOGNITION AGENT. 59
TABLE 4.6: SAMPLE PLAN LIBRARY FOR A PLAN RECOGNITION AGENT IN THE BOX PUSHING EXAMPLE. 67
TABLE 4.7: SAMPLE OBSERVATIONS MADE BY A PLAN RECOGNITION AGENT IN THE BOX PUSHING EXAMPLE. 67
TABLE 5.1: OBSTACLES REMOVED BY INTENT RECOGNITION AGENTS BY DATASET. 74
TABLE 6.1: WEIGHTS USED BY COWS. 78
TABLE 6.2: PLAN LIBRARY FOR AGENTS IN THE COW HERDING SCENARIO. 79
TABLE 6.3: EXPERIMENT SERIES 2 DATASET CONFIGURATION. 82
TABLE 7.1: COMPARISON OF TEAM SCORES BASED ON THE NUMBER OF RECOGNITION AGENTS ON EACH TEAM. 89
TABLE 7.2: COMPARISON OF TEAM SCORES BASED ON THE PLAN THAT THE OBSERVED AGENTS WERE FOLLOWING. 90
TABLE 7.3: COMPARISON OF TEAM SCORES BASED ON THE TIME ALLOTMENT FOR RECOGNITION. 99
TABLE 8.1: BOX PUSHING HYPOTHESIS TESTING RESULTS. 104
TABLE 8.2: COW HERDING HYPOTHESIS TESTING RESULTS. 105
TABLE 0.1: EXPERIMENT SERIES 1. REGRESSION RESULTS FOR TIME AND RECOGNITION TYPE. COMPARES NO RECOGNITION, PLAN RECOGNITION, AND INTENT RECOGNITION. 110
TABLE 0.2: EXPERIMENT SERIES 1. REGRESSION RESULTS FOR DEPENDENT VARIABLE TIME. INDEPENDENT VARIABLE IS RECOGNITION TYPE, COMPARING INTENT RECOGNITION AND PLAN RECOGNITION. 110
TABLE 0.3: EXPERIMENT SERIES 1. REGRESSION FOR DEPENDENT VARIABLE PERCENT. ALL RECOGNITION TYPES REPRESENTED. 111
TABLE 0.4: EXPERIMENT SERIES 1. REGRESSION RESULTS FOR DEPENDENT VARIABLE PERCENT. INTENT RECOGNITION AND PLAN RECOGNITION ARE REPRESENTED. 111
TABLE 0.5: EXPERIMENT SERIES 1. AVERAGE TIMES ACROSS ALL DATASETS BY RECOGNITION TYPE AND POPULATION SIZE. 112
TABLE 0.6: EXPERIMENT SERIES 1: RECOGNITION CHOICES BY INTENT RECOGNITION AGENTS. 113
TABLE 0.7: EXPERIMENT SERIES 2. T-TEST FOR DIFFERENCE IN MEANS OF INTENT RECOGNITION VS. PLAN RECOGNITION. 114
TABLE 0.8: EXPERIMENT SERIES 2. T-TEST FOR DIFFERENCE IN MEANS OF PLAN RECOGNITION VS. NO RECOGNITION. 114

TABLE 0.9: EXPERIMENT SERIES 2. T-TEST FOR DIFFERENCE IN MEANS OF INTENT RECOGNITION VS. NO RECOGNITION. 115
TABLE 0.10: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION. 5 RECOGNITION AGENTS. 115
TABLE 0.11: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION. 5 RECOGNITION AGENTS. 116
TABLE 0.12: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION. 5 RECOGNITION AGENTS. 116
TABLE 0.13: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION. 10 RECOGNITION AGENTS. 117
TABLE 0.14: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION. 10 RECOGNITION AGENTS. 117
TABLE 0.15: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION. 10 RECOGNITION AGENTS. 118
TABLE 0.16: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION. 15 RECOGNITION AGENTS. 118
TABLE 0.17: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION. 15 RECOGNITION AGENTS. 119
TABLE 0.18: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION. 15 RECOGNITION AGENTS. 119
TABLE 0.19: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 2. 120
TABLE 0.20: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 2. 120
TABLE 0.21: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 2. 121
TABLE 0.22: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 4. 121
TABLE 0.23: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 4. 122
TABLE 0.24: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING PLAN 4. 122
TABLE 0.25: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING A RANDOM PLAN. 123

TABLE 0.26: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING A RANDOM PLAN. 123
TABLE 0.27: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION WHEN OBSERVED AGENTS WERE FOLLOWING A RANDOM PLAN. 124
TABLE 0.28: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION. TIME TO COMMUNICATION 50. 124
TABLE 0.29: EXPERIMENT SERIES 2. T-TEST FOR PLAN RECOGNITION VS. NO RECOGNITION. TIME TO COMMUNICATION 50. 125
TABLE 0.30: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS NO RECOGNITION. TIME TO COMMUNICATION 50. 125
TABLE 0.31: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. PLAN RECOGNITION. TIME TO COMMUNICATION 200. 126
TABLE 0.32: EXPERIMENT SERIES 2. T-TEST FOR NO RECOGNITION VS. PLAN RECOGNITION. TIME TO COMMUNICATION 200. 126
TABLE 0.33: EXPERIMENT SERIES 2. T-TEST FOR INTENT RECOGNITION VS. NO RECOGNITION. TIME TO COMMUNICATION 200. 127
TABLE 0.34: EXPERIMENT SERIES 2. T-TEST FOR ONE ADDITIONAL AGENT ACROSS ALL ADDITIONAL AGENT TYPES. 127
TABLE 0.35: EXPERIMENT SERIES 2. T-TEST FOR ONE ADDITIONAL INTENT RECOGNITION AGENT. 128
TABLE 0.36: EXPERIMENT SERIES 2. T-TEST FOR ONE ADDITIONAL PLAN RECOGNITION AGENT. 128
TABLE 0.37: EXPERIMENT SERIES 2. T-TEST FOR ONE ADDITIONAL NON-RECOGNITION AGENT. 129
TABLE 0.38: EXPERIMENT SERIES 2. T-TEST FOR FIVE ADDITIONAL AGENTS ACROSS ALL ADDITIONAL AGENT TYPES. 129
TABLE 0.39: EXPERIMENT SERIES 2. T-TEST FOR FIVE ADDITIONAL INTENT RECOGNITION AGENTS. 130
TABLE 0.40: EXPERIMENT SERIES 2. T-TEST FOR FIVE ADDITIONAL PLAN RECOGNITION AGENTS. 130
TABLE 0.41: EXPERIMENT SERIES 2. T-TEST FOR FIVE ADDITIONAL NON-RECOGNITION AGENTS. 131

Chapter 1 Introduction

In a multi-agent environment, agents complete tasks which are assigned in real time or scheduled ahead of time. However, there may be times when an agent may be idle, such as when it has finished its assigned tasks. An example of this is an agent who has tasks assigned to it at 5PM and 7PM. Between those times, the agent has time when it is idle. If instead of remaining idle, this agent assisted another agent in the system during this time, then the task of the other agent would take a shorter amount of time, thus increasing the efficiency of the overall multi-agent system. As an example of a context where this may happen, one can consider the case of robots whose job it is to stack boxes. Once a robot has finished stacking its boxes, it will sit idly until assigned another task. A more effective system would have agents that recognize that they have idle time that can be utilized to assign themselves additional tasks. For example, in the above scenario, the robot which has finished stacking boxes can then assist other robots in stacking their boxes. Hence, all of the boxes would be stacked in a shorter amount of time leading to a more efficient system.

A key component to a multi-agent system is the mechanism which allows agents to interact. In some cases, the agents may be adversaries and pursuing their own individual or team goals. On the other hand, in a cooperative environment, agents may be trying to achieve an increase in the overall utility of the system. The much researched concept of plan recognition can be used in either of these situations. In plan recognition, an agent attempts to determine the plan that another agent is following. A plan is a set of actions which lead to a result. Plan recognition is used in situations where communication between agents is to be avoided. An example of this is an operation where communications may be intercepted by an enemy agent.

Expanding on the idea of plan recognition in a cooperative environment, this research proposes a novel concept called *intent recognition* (Ahmad and Agah, 2013). In plan recognition, an agent observes another agent in order to determine what plan, or set of actions, that agent is currently following. The goal of an agent performing intent recognition is to aid the other agents in the system. Intent recognition refers to the case where an agent recognizes what another agent *is trying to achieve*, instead of merely predicting the specific steps that the observed agent is following. The agent does this by attempting to follow a plan that has the same intent, or overall goal, as the other agents in the system. Similar to plan recognition, intent recognition aims to minimize the amount of communication between agents. Agents that can perform intent recognition are able to autonomously determine which tasks *they should be performing* in their idle time, in order to increase the overall utility of the multi-agent system.

An example of a scenario is that of a man carrying a heavy bag into an apartment building. His plan is to walk to the door of the apartment building, place the bag on the floor, unlock the door with his key, open the door, prop the door open with something, pick up the bag, walk through the door, place the bag down again, and finally shut the door. This plan would take less time if the man's acquaintance was inside the apartment building and saw him coming through the window. The acquaintance could open the door for the man. The man's new plan would be to walk to the door of the apartment building and walk through the door. Because of the assistance of the man's acquaintance, the overall task time and the number of plan steps would be reduced. Another aspect to consider is that there was no communication in this case. Perhaps the man carrying the bag did not have a free hand to signal to his acquaintance or the acquaintance could not hear the man through the door. In either case, communication was not possible or necessary for intent recognition to take place.

The significance of this research is that it expands the notion of plan recognition to incorporate a new construct, namely intent recognition. It is hypothesized that systems with intent recognition would perform “better” under certain pre-specified conditions. For example, more tasks can be accomplished in the same amount of time. As an example, if an agent is pushing a box towards a specified location and it has obstacles in its path, an idle agent that is utilizing intent recognition may then determine that removing all obstacles between the box and the final location would speed up the completion of that task. It would then perform the actions required to assist the other agent, thus decreasing the amount of time required to complete the task.

Another advantage of intent recognition is a reduction in the amount of communication needed in order for agents to work together. At any given time, a large percentage of agents in a system can be idle. If all of these agents filled the communication channels with requests for information, communication that is vital to the system may not reach the recipient in a timely manner. Reducing communication results in the reduction of all associated costs.

In order to study intent recognition, we built a framework of a multi-agent system in which agents are able to recognize the intent of other agents and utilize their own idle time to assist other agents. This research shows that under certain conditions, an intent recognition system is more efficient than a plan recognition system.

The primary contribution of this research is the introduction and testing of the novel concept of intent recognition. Although the construct of intent recognition has been broached in the literature, as far as we could find, it has not been researched or properly defined. This research defines intent recognition, develops a model of intent recognition, implements this model via simulation experiments, and evaluates the experimental results.

In this work, we present a review of the literature, develop and build a conceptual framework for an intent recognition system, describe the methodology that is utilized and the details of the experimental design, and then discuss the results of the experimentation. The experiments are conducted in two domains, box pushing and cow herding. We conclude with a summary of the significance and implications of this research.

This work addresses the following key research questions: (1) What are intent recognition systems? (2) How can they be used to have agents autonomously assist each other in an effective manner?

1.1 Motivation

In a multi-agent system, agents perform tasks to manipulate or investigate their environment. Agents may work together in a cooperative environment or against each other in an adversarial environment. In either of these cases, there are situations where a reduction in the amount of communication may be beneficial. For example, in an adversarial environment, communications may be intercepted. In the example of a cooperative multi-agent system, communication may be too expensive of a use of resources, such as agents which are located in space. Communication may take place between agents or between a human and an agent. Agents in a multi-agent system that can perform recognition do not need to communicate as often.

We propose the construct of intent recognition. An agent performing intent recognition requires no human intervention when idle to continue completing tasks. Another advantage of intent recognition is that agents can use their unique capabilities to achieve the team goal. Agents may utilize differing plans to work together towards the same goal.

1.2 Research Hypotheses

The first series of experiments - box pushing - has two domain specific hypotheses:

H1.1: Intent recognition systems will be able to reduce task completion time when compared with plan recognition systems.

H2.1: Intent recognition systems will be able to increase percent of task completion when compared with plan recognition systems.

The second series of experiments - cow herding - has two domain specific hypotheses:

H1.2: Teams which include agents using intent recognition systems will be able to score more overall points than teams which include agents using plan recognition systems.

H2.2: Teams which include agents using intent recognition systems will win a greater number of games than teams which include agents using plan recognition systems.

This work has two common hypotheses, for both box pushing and cow herding:

H3: Intent recognition systems will use unobserved actions to complete the given task.

H4: Intent recognition systems will communicate fewer times compared with plan recognition systems.

1.3 Experiment Series 1: Box Pushing

In the first series of experiments, a box pushing scenario is used. Agents pushing boxes are observed by agents performing either plan recognition or intent recognition.

For the box pushing experiments series, we will use the following metrics to evaluate the intent recognition approach:

1. *Time* – The time at which the task was completed.

2. *Percent Completion* – The percentage of the task that was completed correctly.
3. *Communication* – The number of agents that communicated during the task.
4. *Accuracy of Intent Recognition* – The number of agents that recognized a task that either aid or hinder completion of the task.

1.4 Experiment Series 2: Cow Herding

In the second set of experiments, we use a cow herding scenario based on the scenario proposed by the Multi Agent Programming Contest (Dix *et al.*, 2010.)

For the cow herding experiments series, we will use the following metrics to evaluate the intent recognition approach:

1. *Score* – The number of cows that were herded into the team corral.
2. *Wins* – The number of times the team was able to outperform another team.
3. *Communication* – The number of agents that communicated during the task.
4. *Accuracy of Intent Recognition* – The number of agents that recognized a task that either aid or hinder completion of the task.

Chapter 2 Background and Related Work

Intent recognition is different from other existing approaches to agent interaction, such as using an Inverse Model. In an Inverse Model, such as the one by Marhasev *et al.* (2006), the state of the world and the goal of the agent are known. Using this information, the steps that the agent will perform are inferred. There are drawbacks to this approach. It assumes that all agents have the same plan library. Also, it is assumed in this model that the goal of the observed agent is already known. This is an assumption that may not always be true, but is a necessary assumption in that research.

Several studies have incorporated plan recognition in multi-agent systems. A plan recognition system is a system where an agent is observed in order to determine what series of actions it is performing. For example, in the work of Huber and Durfee (1995), agents must observe other agents in order to predict their destination. This approach has the same drawback as using an inverse model in that it is assumed that all agents have the same plans stored in their plan library. This approach does not account for the fact that some agents may have a "better" way of solving a problem than others. The definition of "better" could range from completing the task faster to using the least amount of system resources.

Another example of agents working together is a Sensor Web, such as one the one developed for NASA (Tsatsoulis *et al.*, 2008), where agents work in coalitions in order to determine if a weather phenomenon is occurring or not. When an agent determines that a phenomenon such as a hurricane may be occurring, it searches for other agents to assist it in the verification of this fact. If the system had intent recognition, idle agents would begin assisting each other autonomously, which would reduce the time agents spend looking for help.

The study of multi-agent systems is a broad and diverse field (Wooldridge, 2002, Mataric, 1995, Shoham and Leyton-Brown, 2009, Ya'akov *et al.*, 2012). The key elements of a multi-agent intent recognition system that we will consider in this research include:

- Plan Recognition
- Modeling Agents
- Plan Representation
- Time
- Communication Decisions
- Intent

2.1 Multi-Agent Systems

A multi-agent system is a group of autonomous agents residing in the same environment. An example of this, given by Shoham and Leyton-Brown (2009), is a sensor network, in which each unit has sensing capabilities and limited processing power. Individually, each sensor builds a local view of what is occurring around them. Working together, these agents can build a global view of the environment.

Aspects of multi-agent systems have been studied in great deal. Software frameworks, such as JADE (Java Agent Development Environment), have been developed solely for the purpose of building agents for a multi-agent systems (Bellifemine *et al.*, 2001). Another organization, Foundation for Intelligent Physical Agents or FIPA (IEEE Foundation for Intelligent Physical Agents, 2012) is the standards organization for agents and multi-agent systems.

In our research, we will expand the reasoning capabilities of agents in a multi-agent system so that they are better able to cooperate and thus increases the utility of the system.

2.2 Plan Recognition

When agents are not assigned specific tasks, there must be a mechanism for them to deduce what their current action should be. In a cooperative environment, an agent typically works with other agents to achieve a common goal or to maximize the utility function of the entire group. One way for an idle agent to determine what tasks other agents in the system are currently performing is by observing them.

If the goal of the other agents is not explicitly known, it must be inferred. When agents try to determine what other agents in the environment are doing, it is referred to as plan recognition. In general, plan recognition systems consist of a plan library which stores a selection of possible plans that an agent in the environment may execute. Observations of either agents or the environment are used to determine which plan is currently being followed.

Another way for an agent to find out what to do is to use an inverse model. In an inverse model, the current state of the world and the goal state are known. The actions that are needed to move the current state to the goal state are the output. An example of a type of inverse model is the system built by Marhasev *et al.* (2006) which built on the idea of Hidden Markov Models. This system was designed to monitor the activities of human subjects instead of agents, but the concepts are similar and the people were simulated. The data collected are the location of the task, the object involved with the activity, and the duration of the activity. For the experimental setup, a simulation was run that involved passengers at an airport. The goal of each passenger was to reach his or her gate. Based on the path they took and the time they spent at each step, it was determined whether they had problems during the check-in process (abnormal path) or not (normal path). This type of inverse model is limiting. In order to reduce the search space, there are only two possible paths for the agents. Also, the goal state of all agents is already known. The problem with this is that if an agent is looking for a task to perform and it encounters another agent in the system, the goal state of the observed agent will not be known.

An advantage of plan recognition is that when it is the exclusive way of interacting with other agents, the agents do not need a common communication medium or language. Huber and Durfee (1995) developed a system where the agents did not communicate at all. In this system, an agent moves in a virtual environment to one of the several possible goals. Another agent observes and must determine which goal this agent is moving to. The observing agent must decide at what point it has enough information to determine which goal is correct so that it can move to that location as well. If an agent waits until the observed agent has reached its goal, the total time for both agents to reach the goal is high. On the other hand, if an agent begins movement before the certainty of the goal is high enough, the agent could be moving towards the wrong goal. The problem with this approach is that when uncertainty is high, an agent must wait a long time before performing a useful action. The study by Huber and Durfee (1995) did not explore the idea that sometimes minimal communication may actually be the better solution. For example, in this case, a simple query such as “which goal are you moving to?” may have been less resource intensive, overall.

There are various reasons to reduce the amount of communication in a system aside from saving system resources. Some examples are for surveillance purposes like in the system by Lin and Hsu (2006), to determine what actions a teammate is performing in a soccer simulation such as Huang *et al.* (2003), or for modeling opponents in games like in the systems by Gal and Pfeffer (2003) and Molineaux *et al.* (2009). Wen *et al.* (2012) explore another reason to reduce the need for communication between agents; which is the situation where communication cannot happen instantaneously due to restraints of the system. The method by which these systems compensate for the lack of communication is discussed in more detail in the following paragraphs.

The same processes that help identify activities of another agent in a multi-agent system can be used to do surveillance of human subjects. Lin and Hsu (2006) developed a system called IPARS (Intelligent Portable Activity Recognition System) which was developed for activity recognition. Since there is no communication between the system and the subject of observation, IPARS has a novel way to help limit the search space of possible activities. In the model of activities, actions are associated with various objects. For example, “brushing teeth” would be associated with “toothbrush.” Other factors that are taken into consideration are the duration of the activity, the time of day, and the location of the person. This was compared to a database which contained information about activities, such as where the activity usually takes place and what time it usually takes place. Although the system worked well, an agent in a multi-agent system needs to determine the plans that other agents are following even if the location has not been encountered before. Also, although some tasks are linked to time of day, such as “turning off alarm clock,” it is not difficult to think of many tasks that are not.

The domain of the paper by Huang *et al.* (2003) is RoboCup, a soccer simulation (Kitano *et al.*, 1997). The researchers use an algorithm that not only tries to learn what plans the other team is following, but also applies those learned techniques to their own team strategies. A structured way of describing a plan is explained. In this scheme, a team plan should contain the list of agents it involves, its starting condition, its goal state, and the body, which is made up of steps. Steps are agent behaviors that have temporal constraints on each other.

Huang *et al.* (2003) state that in order to recognize a plan, an agent must be able to take its view of the world (in this case, the ball location, velocity, the angle of the other agents with respect to the ball, etc.) and translate this into what is referred to as element behaviors (such as pass, dribble, shoot). By comparing world views at different time steps, the element behaviors of other agents can be determined by the observing agent. A structured definition of a basic behavior is also described by them as follows:

```
basic-behavior {
    type;
    leading-agent; associate-agent;
    source-region; goal-region;
}
```

Here, the leading-agent is the agent that is performing the behavior, the associate-agent is the object of the behavior, the source-region is where the ball starts, and the goal-region is where the ball ends up. The definitions of source-region and goal-region are domain-specific, but the idea can be applied to other domains as well. The researchers assume that a plan of rational and organized multi-agent team has only one leading action per time interval. A behavior is not confirmed until it has finished occurring.

The system by Huang *et al.* (2003) incorporates several events which activates the recognition function. They are: Control-Ball(agent, t): which means the agent starts controlling the ball at time t; With-Ball(agent, t): the agent continues to control the ball at time t; Release-Ball(agent, t): the agent releases the ball at time t; and Moved(agent, t1, t2): the agent’s position changes between t1 and t2. Using these triggers and behaviors, a two dimensional behavior vector is generated.

Once the vector is generated, the sequences are analyzed and "interesting" and frequent behavior sequences are studied because the agent can then assume that the opposing team's plan is embedded in those sequences. The work uses statistical dependency testing to find trends in the data. A trie is used to store the interesting sequences. Each node (excluding the root node) represents a step. Each node also contains a "count" attribute to show how many times that sequence (or sub-sequence) has occurred. The significance of sequences is then computed with the help of this adapted trie using the Chi-square statistic test (Tabachnick *et al.*, 2001). The Chi-square test evaluates the dependence between the prefix of the sequence and the following step through a test of independency. Further details on the Chi-square test used in that paper are included in Huang *et al.* (2003). Once this is done, the sequences which are deemed important can be converted into the formal plan format. A pseudo-code version of this recursive algorithm is provided in that paper. In our research, we are also interested in the analysis and components of behaviors performed by agents. We refer to behaviors as actions and they are the building blocks of the agents' plans.

In the work of Wen *et al.* (2012), agents are mobile and can only share information with other agents that are in close proximity. Some of the agents are only able to relay information at what is referred to as "disconnected time intervals." Wen *et al.* (2012) used tools from algebraic graph theory and control theory to analyze ways in which agents with these restrictions could reach a consensus. Their research concluded that if the algebraic connectivity of the communication topology is over a threshold value and there are a sufficient number of properly aligned time intervals, then a consensus among the agents can be reached.

Model building is an important aspect of plan recognition, such as in the work of Avrahami-Zilberbrand and Kaminka (2006). When building the model of another agent's plan, an agent will build the possible plan based on observations made of that agent. One way to represent plans in the plan library is a directed acyclic graph (DAG). One way that the DAG can be designed is to have the vertices represent plan steps, vertical edges represent sub-steps, and sequential edges represent the order of execution. The organization of the DAG can be altered depending on the experiment. Their research attempts to address the following issues with respect to plan recognition: handling observations where a component may be lost or missing; dealing with plan execution duration constraints; and interleaved plans (where an agent interrupts a plan for another, and return to the first later). The approaches are built upon algorithms by Avrahami-Zilberbrand and Kaminka (2006) and Avrahami-Zilberbrand *et al.* (2005) because in their opinion, they were the fastest symbolic plan recognition algorithms at the time the work was done. The reasoning for the hybrid symbolic-probabilistic plan recognizer is to use the symbolic component to reduce complexity and the probabilistic component to rank hypotheses, therefore combining the strengths of both techniques. The symbolic component is used to address the aspects of managing issues, interleaved plans, lossy features, and missing observations. Although the details of how the system deals with these issues are not fully explained, it is interesting to note the issues that occur. Although interleaved plans, lossy features, and missing observations are factors in this paper, they are not covered in this research. However, the paper by Avrahami-Zilberbrand *et al.* (2005) also mentioned managing durations, which does apply to our research.

By “managing durations,” Avrahami-Zilberbrand *et al.* (2005) state that instances of the same plan step can vary in their duration. The example given in the paper is “in an airport terminal, there exist a difference in the plans of a passenger who stands at the check-in area for a few minutes, and a passenger who is held there for half an hour.” (Avrahami-Zilberbrand *et al.*, 2005). This issue is handled by the original symbolic algorithm of their proposal and is similar to the temporal issues discussed previously. The idea of managing durations also applies to our research in terms of the issue of time, which will be discussed further in a later section.

The research of Ya'akov *et al.* (2012) states that plan recognition in exploratory domains is NP-hard. That research looked at the recognition of people who are learning to use a statistical software package. While learning how to use the software, users executed exploratory and error-prone behaviors. Users sometimes elect to explore multiple plans at the same time and may interweave plans. In order to manage the complexity, the algorithms proposed by that research are post-hoc. This means that the recognition is done after all the steps have been observed and not after each observation. Plans were inferred by comparing the completed observations with ideal sequences of steps. Our research focuses on domains in which intent recognition can be executed in real time and in which agents perform one plan at a time.

One way in which our research moves beyond the idea of plan recognition, such as the systems by Huber and Durfee (1995) and Avrahami-Zilberbrand *et al.* (2005) is that all agents do not have to have the same plan library or capabilities. The observing agent will be able to determine the intent of the other agents. This observing agent may know of plan steps that are beneficial to the recognized intent, but are not in the plan library of the observed agent. In this way, the observing agent can use new and different capabilities to assist the observed agent. In fact, in our system, it is not necessary for the observed agent to recognize that another agent is helping it.

2.3 Modeling Agents

When playing a game against an opponent, knowing what the opponent intends to do is obviously advantageous. Communication is not an option in this case because intelligent opponents will not willingly reveal their strategies. The work by Gal and Pfeffer (2003) uses

Networks of Influence Diagrams (NIDs) to model the opponent. When playing most games, it is easy to calculate the probability that a given move will result in a favorable effect. However, if a human opponent is modeled to always select the move with the highest probable payoff (called a rational agent), then the model will usually fail. Many times, people use heuristics not directly connected to the actual game rules (referred to as boundedly rational agents), and Gal and Pfeffer (2003) include that in the model of the opponent.

Gal and Pfeffer (2003) define NIDs as a collection of Influence Diagrams (IDs). Each ID is made up of chance nodes, decision modes and value nodes. Edges leading to chance and value nodes represent probabilistic dependence, while edges leading into decision nodes represent information that is available to the agent at the time of the decision. To solve an ID means to compute an optimal strategy for the agent. In other words, what the agent should do, given the available information. If agents have differing views of the world, they may have different values for parameters, even if the structure of their IDs is the same. In a Multi-Agent Influence Diagram, or MAID, each decision node is associated with a particular agent. A NID is a rooted directed acyclic graph where each node is a MAID and the root represents the real world from the modeler's point of view.

In Gal and Pfeffer's (2003) study, during every round of the game, the modeler learns more about the opponent's beliefs and updates the NID. One novel idea is that the modeler has to recognize a variable in order to include it in its opponent's models.

One drawback is that this type of modeling is extremely complex even in a game with simple rules, and the paper introduced a model that they built for a Rock-Paper-Scissors playing agent. One aspect that adds to complexity is that the system has to account for the fact that the opponent may not always be following the same strategy. A variable was introduced to represent the belief of which strategy the opponent was using. Although this system saves resources by reducing the amount of communication, the modeling done to overcome this is far too complex for all but the smallest multi-agent systems.

In order to reduce complexity and develop a parsimonious system, we use Repast Simphony (Repast, 2012) to model our agents. This system is designed to study complex multi-agent systems. Any limitations on agent population and complexity are based solely on local hardware constraints such as RAM size, processor speed, and disk I/O read and write speeds.

In order to determine which plan another agent is currently executing, several possible plans may be under consideration at any given time. An example of a plan recognition system that works in this manner is the Probabilistic Hostile Agent Task Tracker (PHATT) (Geib and Harp, 2004). This system uses a commonly used idea in plan recognition, which is to have hierarchical plans, represented by trees, stored in a library. This system has three steps at every iteration: computation of the complete set of possible plans that an agent may be executing (called "explanations"), computation of the probability that each of those plans is the current plan, and computation of the conditional probability of the possible goal states based on the probabilities of the possible plans. An explanation is a forest of plan instances which is associated with the observed actions of the agent and pending steps associated with timesteps. The pending steps are steps that will occur if a given plan is being executed. Although experiments were run and successfully analyzed, there was one extremely limiting factor: all actions in the plan libraries were unique. According to the work, "once an action is observed there is actually no ambiguity about what root intention the action must contribute to." (Geib and Harp, 2004). There is no indication of how the system would be able to scale in a real multi-agent system where it is unlikely that this condition will hold. We propose a system that can better replicate real multi-agent systems by allowing actions to be repeated in various plans.

Plan recognition can be used to improve the performance of other tasks. For example, Molineaux *et al.* (2009) use plan recognition to aid in the task of learning agents. The process of learning is augmented because plan recognition is used to reduce the dimensionality of the state space. This is because the goal of the opposing team is used as a variable in the reinforcement learning algorithm. In their work, plan recognition is simplified because they already know the opposing team's goal and there are only a few plans to choose from. The directional movements of each player are used to determine the plan. Similar to Molineaux *et al.* (2009), we find the goal, which we refer to as intent, to be an extremely important factor in plan selection. However, in our work, the intent will not be predetermined and will be recognized by the agents.

Another factor that is sometimes considered when performing plan recognition is whether or not all of the agents in the environment have the same beliefs about the observables in the environment. In other words, agents may not have the same values for information which represents the current global state. Even if the facts are the same, the agents' interpretation of these facts may vary. In the plan recognition system developed by McEleney and O'Hare (2004), this idea that different agents may have differing beliefs of the world was explored. At each step of their process, an agent takes a turn. Each agent has a nested belief model which has information about the domain, its own current plan rules, and beliefs about other agents. This model accounts for the fact that the other agents may have different beliefs of the world state. The information is then used to build a decision tree which has choice nodes whenever the agent re-plans. It is assumed that each agent will try to maximize payoff and chance nodes are used to model the fact that the re-planning step is probabilistically known. This results in a cycle of plan-recognition, re-planning, and execution.

The example given in McEleney and O'Hare (2004) was for a car repair task and was in the natural language dialogue domain. The first agent needs a spanner so it says “pass the spanner.” The second agent uses this to determine that passing the spanner should be added to the plan. The first agent then says “Thank you.” The main idea being proposed is that an agent can recognize another agent’s plan and then cooperatively add to it.

The idea of building decision trees to model other agents' beliefs is used in many plan recognition systems. This modeling of both plans and agents is an important concept in plan recognition such as in the previously described work by Gal and Pfeffer (2003), which used the idea of Networks of Influence Diagrams (NID) in the context of games. In the NID, there is a model of the game which is being played along with models of the agents which are playing the game.

Another strategy is for an agent to use its own state as the model for the states of other agents, such as in the research by Jordan and Walker (1996). Their simulation experiments used the Design World Testbed where two agents negotiate on the design of a floor plan of a house. In this system, agents must decide whether or not to remind another agent about information. Although the information shared is already known by all agents, the reminder brings certain data to the current attentional state. Attentional state is modeled after human working memory. In order to model the attentional state of the other agent, an agent uses its own attentional state as a template. Although this does reduce processing time, there can be problems with this approach, since there is no guarantee that the states are indeed the same. In a multi-agent system where agents encounter other previously unknown agents, the chance that the two agents’ internal models will be the same is low. In fact, the novel approach used by our research does not require agents to have the same plan libraries or capabilities.

2.4 Plan Representation

There are many ways to represent a plan. A plan is a list of actions followed by an agent in order to reach a goal. In most cases, there are general actions that an agent may be executing. An example may be "cooking food." Since these actions may be difficult for an agent to recognize as a whole, the activity is broken up into smaller sub-steps. In this case, the sub-steps may include "is located in kitchen" and "opening jar." These actions and sub-steps are referred to by various names. In most cases, an activity is comprised of a sequence of easily recognizable actions. The choice of how to represent plans can be important to the recognition process. Examples of this concept are explored in the works by Lin and Hsu (2006), Huang *et al.* (2003), Hongeng and Nevatia (2001), and Goldman *et al.* (2010).

The IPARS system was discussed in Lin and Hsu (2006), and it was stated that previous work has focused only on recognizing what they call "low level activities." The IPARS system is designed to be able to recognize both high level activities (such as "grooming" and "eating") and low level activities (such as "brushing teeth" and "making tea"). The paper also refers to high level activities simply as "activities," while low level activities are referred to as "actions." According to Lin and Hsu (2006), "actions are used to enhance the accuracy of activity recognition." These activities can occur in parallel or in sequence depending on the plan.

Huang *et al.* (2003) also used the concept of high level and low level activities, though using a different terminology. This was the work described earlier which implemented agents in the RoboCup domain. The work defines element behaviors as "the minimal cooperative behaviors between agents." An element behavior is the smallest action which still conveys meaning to another agent. Examples of these are “pass”, “dribble”, and “shoot”. These elements are further broken down into “actions” or “events” which can be recognized by making an observation at a single timestep or comparing observations of multiple timesteps. Examples of these are “Control-Ball,” “Release-Ball,” and “Moved.” The actions and element-behaviors are used to determine which team plan an agent is following.

Following along the same lines as the previously discussed research, the work by Honeng and Nevatia (2001) classifies possible agent events as “complex” and “simple” events. The main focus of their research is to determine how an event is recognized. Although the experiments had to do with events viewed from a camera, it brings up several ideas which are relevant to other multi-agent systems as well. An issue raised in their study is that events should be broken up into easy to recognize “single thread” events. For example, “converse” can be split into consecutive simple events such as “a person approaches the reference person” and “stops at that person.” This work uses a finite state automaton to represent a complex event. Complex events need logical and time relations between them, an important concept which will be discussed in more detail.

In this research, we use the two level approach to classifying actions described above. We refer to them as “complex” and “simple” actions in the same way as Honeng and Nevatia (2001).

Something that makes plan recognition complex is the number of plans in the library. When observing an agent, if the list of possible plans is not reduced, the problem of plan recognition can become very complex. According to Geib and Harp (2004), the developers of the previously described PHATT system, maintaining and analyzing a long list of possible plans is costly. Because of this, either the plan library should be small or the search space should be reduced in some manner. PHATT does this by having all actions be specific to a plan, so that no action can occur in more than one plan. However, this is not a realistic solution. In our research, we allow an action to occur in any number of plans.

The method by which Huber and Durfee (1995) reduced the complexity of plan recognition was by starting all agents with the same plan. They then make observations about each other to see whether or not the plan is still being followed. The overall goal of the system is for agents to determine when to drop a team commitment based on the observed commitment levels of the other agents. This strategy only works when all agents know the team plan at the beginning of the simulation. It leaves no room for switching to a new team plan, only dropping the current one. Also, the research is not practical for situations where the initial plan or goal of the other agents is not known.

One variable that is not used in most plan recognition systems is “negative information.” These are observations about what an observed agent did not do. Gabaldon (2009) uses this to solve a problem that most plan recognition systems have, namely that every previous action of the observed agent has to be witnessed. Also, it addresses the issue of having to observe all the actions in the correct order. Using negative information, Gabaldon (2009) addresses the issue that some systems do not allow actions to be shared between plans.

The study by Gabaldon (2009) represents an activity as a pair consisting of the name of an activity and a sequence of actions. An action can be another activity. Similar to other plan recognition research, all observed actions are assumed to be deliberate and working towards a plan.

In our research, Gabaldon (2009)’s idea of negative-information is used in the calculation of how likely a particular plan is to have the same intent as the observations. Similar to that study, our research is able to share actions between plans.

Even though plan recognition is usually conducted in multi-agent systems, it is usually performed on one observed agent at a time. Banerjee *et al.* (2010) focus on multi-agent plan recognition where a set of agents, or team, has behaviors that are observed. Instead of having one or more libraries of single agent plans, their research uses team plans in the plan library. In order to achieve this, one assumption that is made is that all single agents behave the same way under the same conditions. To represent the plans, the researches transform Hierarchical Task Network (or HTN) plans into strings which they call a "flat representation" of the plan. Each string is made up of symbols which represent the path through the graph.

The research of Komenda *et al.* (2013) focuses on cooperative agents that share the same goals. In particular, they look at the case where a plan that the agents are following fails. They compare the scenario where agents create a new plan from scratch to the scenario where agents build a plan from a previous plan. Komenda *et al.* (2013) refer to this as "repairing" a plan. In our research, agents are able to select from predefined plans from their plan library. If an agent is unable to complete the plan or reach the goal, it begins the recognition process again in order to select another plan from the plan library.

The research of Goldman *et al.* (2010) broaches the question of how to determine when two plans are equivalent. A contributing factor to this topic is the way in which a plan is modeled. There are instances where a plan may seem equivalent when they in fact have differences between them. For example, considering plan steps *a*, *b*, *c*, and *d*, there could be a plan whose sequence is *abcd*. There could also be a plan which consists of the sub-plans *ab* and *cd*. Both of these plans would be "language equivalent" and could be written as *abcd*. One possibility is to model the probability of each plan step, as well. Goldman *et al.* (2010) state that the choice in how to model plan libraries needs to be carefully selected. We agree with this conclusion and incorporate that in our research.

2.5 Time

When events are observed, they can be given temporal relationships. A single-agent or multi-agent belief network can then be constructed to reflect the temporal nature of the observed actions. These can then be matched against plans in the plan library.

The example used in the work by Intille and Bobick (1999) is a football game where the play is determined using observed events. When events are observed, time is conceptualized by temporal relationships such as "before" and "after." A multi-agent belief network, which is similar to a naïve Bayesian classifier (Rish, 2001), is constructed to reflect the temporal structure of the action. This network represents a particular play using beliefs and evidence about the expected temporal relationships between agent goals in that play. In this case, "agent goals" refer to the specific task that the agent was performing in context of the given play. The network has two types of nodes, observable evidence nodes and unobservable belief nodes. Observable evidence relates to events that can be directly observed. The unobservable belief nodes can either be true or false. They represent some state of the world or agent. Temporal relationships between agents are linked directly to the top-level belief node. Their work stresses the importance of the issue of time in regard to plan recognition.

However, Lin and Hsu (2006) actually measure the duration of activities and this is a key factor in their system. On the one hand, a human may be interacting with an object, but the duration of the activity is so small that it may have occurred by accident, such as a person accidentally brushing against furniture while walking. Yet, in other cases, the activity itself is very brief, such as taking medicine, but is important and should not be ignored.

Huang *et al.* (2003) discuss the idea of temporal constraints. They claim that typical temporal constraint networks which have asynchronous restrictions between agents' behaviors are not accurate representations of team plans. Instead, they state that their synchronous representation is simpler, but is equally effective. According to Huang *et al.* (2003), a plan consists of multiple steps, but each step can contain the co-occurring behavior of multiple agents.

In our research, time will be used as one indicator to evaluate how efficiently a task is completed. If an agent spends a long time for completing a particular task, other agents in the system may need to assist it to complete that task.

2.6 Communication Decisions

The idea of reducing the amount of communication in a system follows along the same lines as Behavioral Implicit Communication (BIC) of Castelfranchi (2006). In that theory, communication is replaced by an agent's behavior. The act of observing an agent and deriving meaning from its actions is called "signification." There are many levels of BIC. In the weakest form, an agent is not acting in order to communicate; and communication is not the intent of the acting agent. In that case, the agent is merely aware that its actions may be observed. In another form of BIC, an action may be necessary, but should also be done for the purpose of communicating. In the strongest form of BIC, an action is done for the sole purpose of communicating. These actions may include a ritual or a simulation of another action. The work done in this subject is rich in theory but is lacking in detailed experimental analysis to test its validity.

One way to determine what is the plan or goal of an agent in the environment is to simply ask them. Communication is an important aspect of a multi-agent system. It is the primary way in which agents interact. There are various reasons that an agent may need to communicate with other agents. For example, agents may not have a global view of the environment and may need to acquire non-local information, such as in the work of Xuan *et al.* (2001). They describe the case where cooperating agents in a multi-agent system are trying to maximize their overall utility. The specific example takes place on a grid-like world where two agents which do not know each other's positions must meet within the given time limit. Unlike other works, the researchers introduce the idea of including communication as a possible action (instead of something that occurs before or after an action), and therefore a cost is associated with it. Their reasoning is that communication also takes up resources and agents should contemplate whether it is appropriate to communicate or not, at any given point. The example used in the study related to grid world; and many variables were manipulated, including the communication cost. The selected policy is understandably more efficient when communication is free. When the communication cost is high enough, the policy has to be changed to compensate for this until a balance is reached. This research brings to light the importance of communication in a multi-agent system.

However, there are various situations in the real world where agents cannot or should not contact other agents. In the previously described work by Huber and Durfee (1995), all decisions were made without communication. The simulation used for their research assumed that the agents were surrounded by hostile agents and were on a military mission. In that situation, communication could bring attention to the agents or be intercepted. Another reason given in their work is that an agent may stop communicating if it is incapacitated. Agent communities that do not rely on communication are still able to function under these circumstances.

Communication strategies are defined in different ways. An example is the research by Mammen and Lesser (1997) who state that there are key points in an agent's problem solving process. These key points include the point after an agent assigns a certain number of variables, or the point when an agent is forced to backtrack. Communication only takes place at these key points. Various communication strategies require an agent to communicate at different key points.

Some researchers argue that minimizing agent communication in multi-agent planning makes it the most efficient. Brigs and Cook (1995) argue that while sharing knowledge is crucial, communication can be extremely costly and therefore each situation should occur with the least amount of communication. An example is given of a factory floor. Each agent in the environment has different capabilities and goals. For example, one of the agents is able to move boxes, and its goal is to move boxes from Room 1 to Room 2. Communication is needed in this case because of a narrow doorway between the rooms that only one agent can fit through at a time. Also, other agents have goals related to the boxes, so that is also a shared resource. Communication is necessary to coordinate the actions and resources of this agent, but communication costs are set to be high so that only the minimum amount of communication is used. To achieve their goals with the least amount of cost, the agents in this particular environment communicated initially to decide on a subgoal for each agent, and then they did not communicate again until they had achieved their respective subgoals. While this works in this specific situation, agents should be able to communicate if there are unexpected circumstances during the simulation experiments. In our research, we take this into account and agents are allowed to communicate in certain circumstances.

A related subject is the concept of coordination through observation, such as in the previously described research by Huber and Durfee (1995), where agents must predict which target was the destination of another agent in the simulation. In their theory, plan recognition is the method by which agents primarily accumulate information. However, instead of studying the appropriate level of communication for the task, the simulation experiments were conducted without communication at all. An interesting extension to this research may have been to see what the optimum level of communication was to make the simulation the most efficient. There are risks and benefits to both communication and plan recognition. It is important to find the situations when it is better to pick one strategy over the other.

In the research of Huber and Durfee (1995), agents are able to reason about the time at which to abandon their commitment to a plan without communicating with the other agents. The agents in this system work with joint (or group) plans. In the experiments, two or more cooperating agents work together in a military reconnaissance mission with the goal being “bounding overwatch,” which is a military tactic. Based on observation alone, it should be obvious to an agent whether the other agents have dropped their commitment to the goal or not. The agents in this system include observations of the other agents’ commitment along with their general observations of the environment. An example of an observation that would reduce the amount of confidence in the goal state of the other agents would be that if an agent was observed while moving into foliage, or hiding. To reduce the complexity of plan recognition, this work dealt solely with de-commitment issues. In other words, the agents’ main objective was to determine whether or not the other agents were committed to the given team plan. The research did successfully show that perception and inference can be used instead of communication.

According to Xuan *et al.* (2000), communication should be an expressed action which has a cost associated with it. In other words, the decision to communicate or not also becomes an explicit action in the agent’s plan. An agent’s policy for communication and coordination is stored in a tuple. The example given was the previously described grid world example. In this simulation, where agents are trying to maximize the total reward with minimal communication, the total reward for executing a plan is calculated as the terminal reward plus the rewards collected at all of the previous steps, minus the communication costs of each of the communicating agents. In order to reduce communication, sometimes generalizations about the simulation have to be made. In this case, agents rely heavily on the assumption that all other agents in the simulation have the same goal. This, however, is very limiting and there are very few situations where this assumption is realistic. In our research, goals of other agents vary.

The previously described paper by Xuan *et al.* (2001) is another example where costs were associated with communication. According to the paper, communication costs in a multi-agent system can relate to various real world costs such as transmission fees and resource costs. In their work, communication cost can be affected by many factors, including the particular time and state. Time is a factor in their simulation experiments because there is a given time limit in which the task must be completed.

While different research efforts have varying methods to evaluate whether the amount of communication was effective or not, the idea of including time as a constraint is seen in many studies. An example is the work by Mammen and Lesser (1997). They developed a "distributed problem-solving testbed" meant for analyzing inter-agent communication, coordination, and problem-solving performance. Communication was explored by manipulating the communication strategy and communication delay. The communication strategy was defined as the point in problem-solving where sub-task results are passed along. Their work proposes that communication delay could be further specified by its mean and variance. In order to evaluate how effective a particular strategy was, the group considered several factors, including the number of undone tasks. However, the main measure of problem solving efficiency is the total system problem-solving time. We also used time and task completion as a measure of efficiency in this research.

Walker and Rambow (1994) built dialog planning agents which existed in the previously described Design World. While the research mainly focused on modeling the attentive state of the agent or the human that would hear the text, this study also placed a great deal of focus on the idea of communication costs. The overall performance of the system was influenced by the cost of sending a message (COMMCOST), the cost of inference (INFCOST), and the cost of retrieval from memory (RETCOST). The formula used was (Walker and Rambow, 1994):

Performance = RAW SCORE – (COMMCOST * total messages) – (INFCOST * total inferences) – (RETCOST * total retrievals)

The study reported that certain strategies which otherwise work effectively become detrimental when communication is expensive. This opens up many interesting research topics such as how an agent selects a strategy based on the current cost of communication.

In McEleney and O'Hare (2004) and McEleney and O'Hare (2005), dialog length was one of the factors to evaluate how well the system performs. Taking this idea one step further, the shortest possible dialog was the overall goal of the system. The “planner,” as it is referred to in the studies, is a system where agents negotiate about a plan which they both will collaboratively execute. Dialog is represented as a game-tree which is constructed using various strategies, all in the pursuit of rewards. Each type of dialog act, for example “ask” or “propose,” had an associated cost which is fixed. The agents tried to minimize cost while planning their dialog strategy.

However, there are a few research efforts that do not focus on minimizing communication. In Jordan and Walker (1996), agents decide when to remind other agents of information that is mutually known. Although the agent already has the data in its knowledge base, a reminder of certain facts helps shift the agent's focus to those facts. This is a case where extra communication is perceived to be beneficial to the task completion. In our research, we consider extra communication as cost to the system and hence it is not seen as beneficial to task completion.

If an agent is attempting to assist other agents while minimizing communication, it needs a way to determine when another agent may need assistance. Although there has not been a lot of research in this area, Marhasev *et al.* (2006) introduced a concept that may be used for this purpose. This is that although two agents may be following the same plan steps, the amount of time spent in each step is important, as well. The provided example was the one of airline passenger checking in for their flight. If the passenger spends a short time at the check-in counter, they are more likely to head straight to the gate. On the other hand, if the passenger spends a long time at the check-in counter, it is more likely that they encountered a problem and will speak to a manager as their next step. Avrahami-Zilberbrand and Kamina (2006) add to this concept. An agent which makes an observation at every time interval T can recognize the duration of other agents' activities. If the agent observes that another agent is performing activity A at time steps T through $T + k$, it can assume that the agent spent at least k time steps at that step of the plan.

Communication uses the resources of both the agent that initiates the process and the agent that is the receiver. If agents are in the process of performing a task, they may have to wait for the dialog to be completed before continuing. In order to limit communication, communication costs are applied not only to the agent that initiates the conversation, but also to the agents which respond. While cost efficient, saving resources by communicating less has a higher risk of incorrectly predicting which immediate task will be most beneficial to the overall goal.

Another reason to not rely on communication is the situation where there are time delays associated with either sending or receiving communications. There has been much research on this topic, including the works by Wu *et al.* (2012) and Tian and Zhang (2012).

As it can be seen from this survey of communication in a multi-agent system, there is an argument for disallowing communication completely. Since this is very severe and communication is an important part of multi-agent systems, communication is allowed in our research. However, it is only used when all other options are exhausted. Since communication has a cost and uses resources, the objective of our research is to minimize communication whenever possible.

2.7 Intent

The subject of intent has been broached in multi-agent systems before, but the research sometimes fails to explain what the definition of intent is. The definition often relies on domain specifics, as is the case in the example of the story generation domain. The meaning and purpose of "intent" in the case of story generation is different than intent recognition in multi-agent systems. An example of this is the work by Riedl and Young (2004). In this domain, intent is thought of as the intent of the character in the story, i.e., what they are trying to achieve in the plot. Intent recognition is performed by the audience. The goal of that research is to achieve "plot coherence" and "character believability." Also, all of the choices made by the individual agents have to make sense together in the story. The research has a "Planner" which simulates the intention recognition done by the audience. In story generation algorithms such as the research by Riedl and Young (2004), intent recognition is defined as determining whether character actions are intentional or not. In multi-agent systems, all actions are intentional and the goal is to determine what that intent is. In other words, in one case, it is trying to determine whether the actions are intentional or not, while in the other case the purpose is to discover the meaning behind each action.

In the field of speech processing, intent can be used to create systems that are more easily used by the consumer. For example, Kanevsky *et al.* (2013) created improved techniques for a speech recognition system to recognize human based grammar. With the system better able to recognize the intent of the speaker, it reduces the need for awkward preprogrammed commands that the speaker must recognize.

The research of Zhang *et al.* (2012) shows how intent can be used to create better prosthetics for humans. They implement a "cyber physical system" for a neural machine interface. The interface senses signals from the human in order to determine the intended motion in real time.

When seen as an expansion of plan recognition in multi-agent systems, intent recognition is a relatively new and unexplored construct. Due to this fact, there is sometimes confusion as to the distinction between the two types of recognition. An example of this can be seen in the work by Bigelow (2013). That work states that intent recognition "is the problem of translating a series of sensor inputs into a prediction of the action that another agent is about to perform (Bigelow, 2013)." However, this definition more readily defines plan recognition, as the future plan steps are being predicted. Intent recognition, on the other hand, is the attempt to recognize what the observed agent is trying to achieve as opposed to the specific steps to reach that goal (Ahmad and Agah, 2013).

In the work of Sadilek and Kautz (2010), intent recognition was attempted in cases where there is a lot of sensor noise or a sensor malfunction. In order for intent recognition to take place, the individual actions or activities of the agents needs to first be recognized. In their work, the capture the flag domain was analyzed. For their research, Sadilek and Kautz (2010), enlisted volunteers to play capture the flag around the University of Rochester. There were two teams of seven people. Each player had a GPS device which logged the player's location, along with noise, every second. The goal of the game was to enter the opposing team's flag area. Players could be captured on enemy territory by being tagged by an opposing player. Once captured, a player was to stay in place until freed by a teammate. GPS accuracy varied from one to ten meters of noise.

Due to the noise, when trying to recognize the activities and intent of a particular player, Sadilek and Kautz (2010) considered the player's relationship with and effect on the other players. The movement of the individual players was also analyzed over long time periods. For example, if a player A approached player B and player B begins to move away quickly, a possible action that occurred was that the two players were on opposing teams and player B was attempting to evade capture. However, this is not the only explanation of what had occurred. Another possibility is that players A and B are on the same team and that they decided to proceed in opposite directions to maximize the team's coverage area. In order to overcome this ambiguity, a large portion of the research by Sadilek and Kautz (2010) focused on the modeling of success and failure. The previous example would have been modeled as a failed attempt to capture an opposing player and a successful attempt at strategizing with a teammate. By modeling these possible successes and failures over time, the intent of the player was then analyzed and interpreted.

As opposed to the research by Sadilek and Kautz (2010), in our research, we directly observe the agents on which recognition is being performed. Also, our agents do not need to be observed for long periods of time. This is advantageous because our agents can receive assistance in real time.

2.8 Other Approaches

Work has been done in the area of human workgroups, which are similar to multi-agent systems, where the coordination takes place between people instead of agents (Hackman, 1990, Majchrzak and Gasser, 1992). It has also been shown that research in human work groups is applicable to robotic communities and multi-agent systems (Agah, 1995). This type of research does not apply to intent recognition because the primary way for humans to convey information to each other is via communication (implicit and/or explicit).

Another approach to multi-agent systems that do not rely on recognition is flocking behavior, where agents use sensory data to follow a virtual leader (Su *et al*., 2009). A variation of this can be seen by the ancillary cow agents in our second experimental series.

In an approach called holonic multi-agent systems (Rodriguez *et al.*, 2011) agents are semi-autonomous. These agents can come together to form a single entity. While the agents in our research are working together, they are completely autonomous.

2.9 Multi-Agent Contest

The Multi-Agent Programming Contest (MAPC) has been held each year since 2005 (Dix *et al.*, 2013). It is currently organized by staff from Utrecht University, Clausthal University of Technology, and Delft University of Technology. The organizers work closely with Computational Logic in Multi-Agent Systems (CLIMA) and International Workshop on Programming Multiagent Systems (PROMAS). There have been several different scenarios for the contest. The following sections describe the various scenarios in the Multi-Agent Programming Contest.

2.9.1 Food Gatherers

In 2005, the contest was a “Food-Gatherers” scenario. In this scenario, the environment is a 20 by 20 grid-like world where each space can be occupied by one agent. There was only one team present on the grid at a time. Each team consists of four agents who initially begin in one of the four corners of the grid. In this environment, food can appear in all of the spaces except for one. This special space is like a team base where the agents can collect their food. When an agent is on a space, it can observe if there is currently food there. Initially, food is distributed in randomly selected locations. During the scenario, additional food appears every 20 seconds in randomly selected locations, except the special space. There are no assigned roles for any of the agents. In this scenario, teams did not directly compete with each other. In our work, our teams directly compete with each other, so we did not focus on the food gatherers scenario.

2.9.2 Goldminers

In 2006 and 2007, the contest was the “Goldminers” scenario. In this scenario, an agent team consists of four agents. Each team tries to get the most gold. The environment is a rectangular grid consisting of cells. The size of the grid is variable with a maximum of 100 by 100 cells. The simulation is run for a finite amount of time.

The cells can contain an obstacle, gold, an agent, the depot (a location where gold items are delivered), or a mark (a string data with a maximum of 5 characters which can be read/written/rewritten/removed by an agent). The locations of obstacles, gold items, and initial agent positions can be either assigned for the particular scenario or random. During the simulation, gold items can appear randomly in empty cells. At the start of each simulation, agents are provided with details of the environment such as grid size and depot position. An agent can perceive the content of the current location and the eight cells surrounding it. If two agents are standing in each other's field of view, they will be able to recognize whether or not one is an enemy. An agent is not able to recognize whether or not the other agent carries gold.

Agents are allowed to perform one action in a simulation step. The list of allowed actions consists of: skip, move east, move north, move west, move south, pick, drop, mark, and unmark. An agent can carry one gold item at a time. The agent can pick up the gold if the cell in which the agent currently stands contains the gold and the agent is not currently carrying another gold item. Dropping a gold item to a depot cell increases the score of the agent's team by one point. An agent is allowed to mark a cell it currently stands in by a string data with a maximum of five characters. The depot cell and cells containing an obstacle cannot be marked. Marked cells can be rewritten with new marks.

An agent carrying a gold item can enter the depot cell and must drop the gold item as the very next action it executes. It must then leave the cell in the first simulation step where it is able to move. If an agent does not meet these conditions, the agent will be moved to a random empty cell.

The team with the most points at the end of the simulation wins. In our research, we also wanted a competition based experiment. However, we did not use the goldminers scenario as we were looking for a domain which lends itself to a greater complexity in terms of team reasoning.

2.9.3 Agents on Mars

In 2011 and 2012, the contest was the “Agents on Mars” scenario. The goal of this scenario is for a team of agents to occupy the best zones on Mars. Zones are rated based on the quality of their water wells. Agents can sabotage rivals and defend themselves. Agents have differing capabilities such as repair or special sensing. Teams must cooperate and coordinate themselves to achieve this goal. In this case, the environment is represented as a graph where vertices denote water wells and possible agent locations. Once a zone is occupied by a team, the team gains the number of points equal to the number of vertices in the zone. The map is unknown at the beginning of the scenario.

A team consists of 20 agents. There are five roles (explorer, sentinel, saboteur, inspector, and repairer) with four agents per role. The explorer agents can find water wells and help explore the map. The sentinel agents have long distance sensors and can observe large areas. The saboteur agents can attack and deactivate enemies. The inspector agents can spy on an opponent’s agents. The repairer agents can restore damaged agents. Every team plays against all of the other teams and each match has several rounds. The team that wins the most rounds wins the overall tournament.

At the time that our research began, the “Agents on Mars” scenario was still fairly new. There were fewer papers written about the topic than the previously existing contest domains. We decided not to utilize “Agents on Mars” for our domain in lieu of a more thoroughly studied domain.

2.9.4 Cows and Cowboys / Cow Herding

From 2008 to 2010, the contest used the "Cows and Cowboys" scenario, which is also known as the "Cow Herding" scenario. In this competition, the environment is a grid-like world in which cow agents are moving around collectively in one or more groups. The size of the grid can vary with a maximum grid size of 150 by 150. There are two corrals, one for each of the agent teams. Each team of agents competes to control the behavior of the cows and lead them to their own corral. The edges of the corral are marked with "obstacle" objects which cannot be moved or traversed by the agents, as shown in Figure 2.1.

Agents are able to recognize whether other agents are from the opposing team or they belong to the same team. The corral positions are identified at the beginning of each match.

Each year had a slightly different variation of the cow herding scenario. One year, the corrals consisted of three sides of a rectangle and were open on one side. Another year, the open side of the corral was blocked by a fence which could be opened by a button. The fence opened when an agent was located in a position which was adjacent. In one of the variations, there were six agents per team while in another variation there were twenty agents per team.

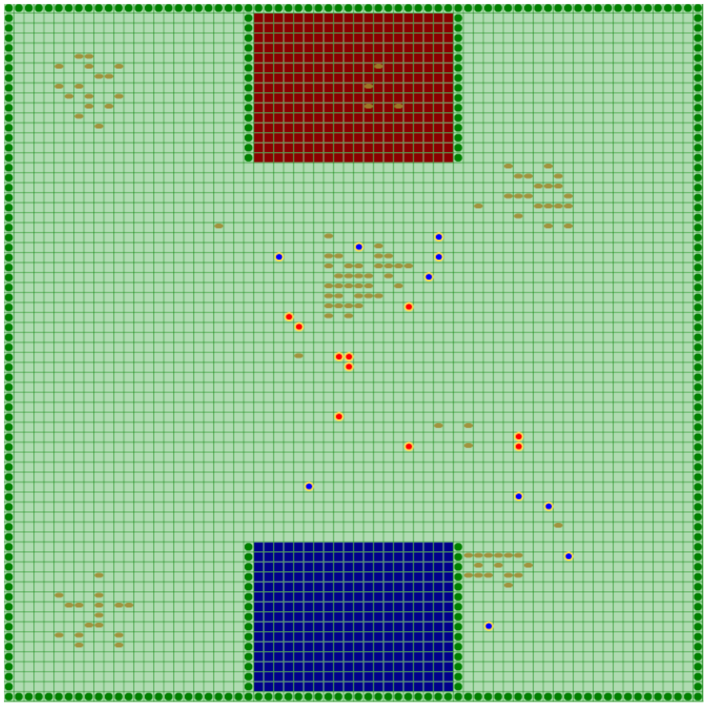


Figure 2.1: Example image from 2008 cow herding scenario document (Dix et al., 2008).

In the first year that the cow herding scenario was used, the score was calculated by the number of cows that were brought into the corral. Once a cow entered the corral, it was not able to leave. After the first year, the score was calculated as the average number of cows in the corral instead. In this new version of the scenario, cows could repeatedly escape from the corral and be herded again until the simulation reached the maximum time limit. To calculate the score for this version, at every timestep, the cows in a team's corral are counted and added to the sum for all of the previous timesteps. The final sum is then divided by the number of steps, which yields the final score for that team.

The cow herding scenario is more complex than the Food Gatherer and Goldminer scenarios. This scenario was successfully used for several years for the MAPC unlike the relatively new Agents on Mars scenario. Due to this, when we began our research, there was a greater body of research about the Cow Herding domain than the Agents on Mars domain, so we selected the more thoroughly studied domain. This cow herding scenario is used for the second set of experiments in our research. Our corrals have three sides and the cows can escape the corral.

2.9.5 Cow Behavior

The cows in the cow herding scenario all follow the same algorithm which can be seen in Figure 2.2 and Figure 2.3 . The cows in our research use this same algorithm.

Algorithm 1 Cow movement algorithm.

Require: a cow represented by its position vector $c \in \mathbb{N} \times \mathbb{N}$

let N be the set of the 9 cells adjacent to c, including c;
remove from N all those cells that are not reachable;
calculate the weights of all cells $n \in N$;
determine the set $M \subseteq N$, where the weight for each $m \in M$ is maximal;
randomly pick a cell $m \in M$;
move the cow to m;

Figure 2.2: Cow Algorithm 1 located in 2010 MAPC scenario document (Dix *et al.*, 2010).

Algorithm 2 Calculate the weight of a given cell.

Require: a cell represented by its position vector $n \in \mathbb{N} \times \mathbb{N}$, and a cow-visibility range $r \in \mathbb{N}$

determine the set C of all cells that are a in the rectangle $[n_x - r, n_y - r + n_x + r, n_y + r]$ and that are on the map;
set ret to 0;
for all $c \in C$ **do**
    calculate d the distance between c and n;
    get the weight w of c in respect to the cell content;
    add w/d to ret;
end for
return ret

Figure 2.3: Cow Algorithm 2 located in MAPC 2010 scenario document (Dix *et al.*, 2010).

2.9.6 Cow Herding Strategies

Boss *et al.*(2009) developed a team for the Cow Herding contest scenario. The team's strategy was to maximize the score as opposed to stopping opposing team from scoring points.

There were three kinds of agents on the team: herders, scouts, and a leader. The leader agent is a herder and it also delegates targets to everyone. Herders are assigned targets based on their current position. Scouts decide where to go on their own. All of the agents calculate paths using A* path finding algorithm (Dechter and Pearl, 1985).

Agents on the team attempted to keep cows together in herds. The teammates had a shared view of the world to avoid the situation in which two agents were herding the same cow. Agents do not deliberately attempt to herd cows that opponents are herding. However, this scenario may occur if the cow happens to be in a location deemed to be ideal for herding.

Similar to the research of Boss *et al.*(2009), our agents will avoid herding cows that are already being herded by a member of the same team. Also, similar to the research of Boss *et al.*(2009), an agent may attempt to herd a cow that is being herded by the opposing team. This will not be intentional aggression, rather it will be inadvertent and due to the location of the cow. In other words, the agents only attempt to discern whether a potential cow target is being herded by a teammate. Cows being herded by an opposing team member are seen in the same manner as cows that are not currently being herded, which can sometimes cause inadvertent adversarial behavior. Another concept that is similar to our research is the concept of a leader agent who assigns targets to the other teammates based on location.

Heßler *et al.* (2010) analyzed their team strategy for the 2009 contest submission. These researchers also competed in the 2007 and 2008 contests. Agents on this team could take on various roles. An agent in the “explorer” role explores the environment, and records the location of cows, obstacles, and the opposing corral. An agent in the “herder” role drives one or more cows to the team corral. An agent in the “keeper” role prevents cows from escaping the corral. Agents in the “opponent analyzer” role analyze opponent behavior and can possibly interfere with their efforts. An agent in the “team analyzer” role analyzes the behavior and performance of its own team. Agents can take on more than one role at a time.

The overall team strategy begins with the agents exploring and locating cows. The agent with the shortest distance to a cow is assigned to herd that cow to the corral. Other agents keep exploring until they find a cow of their own. All agents calculate their paths using the A* algorithm. This team was a contest winner. A simplified version of this strategy is used as a possible observed agent team strategy. It is used as an example of an efficient team strategy.

The research by Heßler *et al.* (2010) explores the concept of agent intentions. These intentions are domain specific and indicate what the agent is trying to achieve next. Examples are exploring and driving a cow. An agent sends its intentions to other teammates so they can utilize the information when making decisions about what their own intention should be. Every agent broadcasts its perceptions and intentions to the other teammates. The strategy used by the research of Heßler *et al.* (2010) leads to many agent communications. Our research uses the idea of agent intentions, which we refer to as an agent’s intent, which results in reducing the amount of communication.

The team by Rahmani *et al.* (2009) also submitted a team for the MAPC. In their team, agents can either explore or herd. An agent’s exploration area is assigned by coordinator. The agents herd one cow at a time, and do not make large herds purposefully. All of the agents are independent. According to authors, there was a lack of effective cooperation between the agents.

The individual agents act appropriately but do not cooperate well in terms of exploring and herding.

In our research, agents are also independent. However, our agents use intent recognition to overcome the issue of lack of cooperation between agents.

2.10 Related Work Summary

As the review of the literature indicates, several frameworks for multi-agent systems have been proposed, such as plan-recognition. However, none of these frameworks define and assess the construct of intent recognition which we propose in this work. Intent recognition is the process of recognizing another agent's objective while minimizing communication. This is different from plan recognition because determining and predicting individual plan steps are no longer the focus of the recognition. This research expands on the idea that agents with differing capabilities can use their individual strengths to work together to solve a common problem. We propose the novel concept that incorporating intent recognition will enhance the effectiveness and efficiency of a multi-agent system. This concept is implemented and evaluated, as we develop a conceptual model based on our intent recognition framework.

Chapter 3 Intent Recognition

The main advantage of intent recognition over plan recognition is the ability to dynamically choose actions to assist other agents in the system (Ahmad and Agah, 2013). An intent recognition agent that is pulling from its plan library may choose an action that it has not observed. Both types of agents, plan recognition and intent recognition, have plan libraries. However, intent recognition agents have the advantage of not needing the same plan libraries as the agents that are being observed. The general recognition structure for both plan recognition and intent recognition is shown in Figure 3.1.

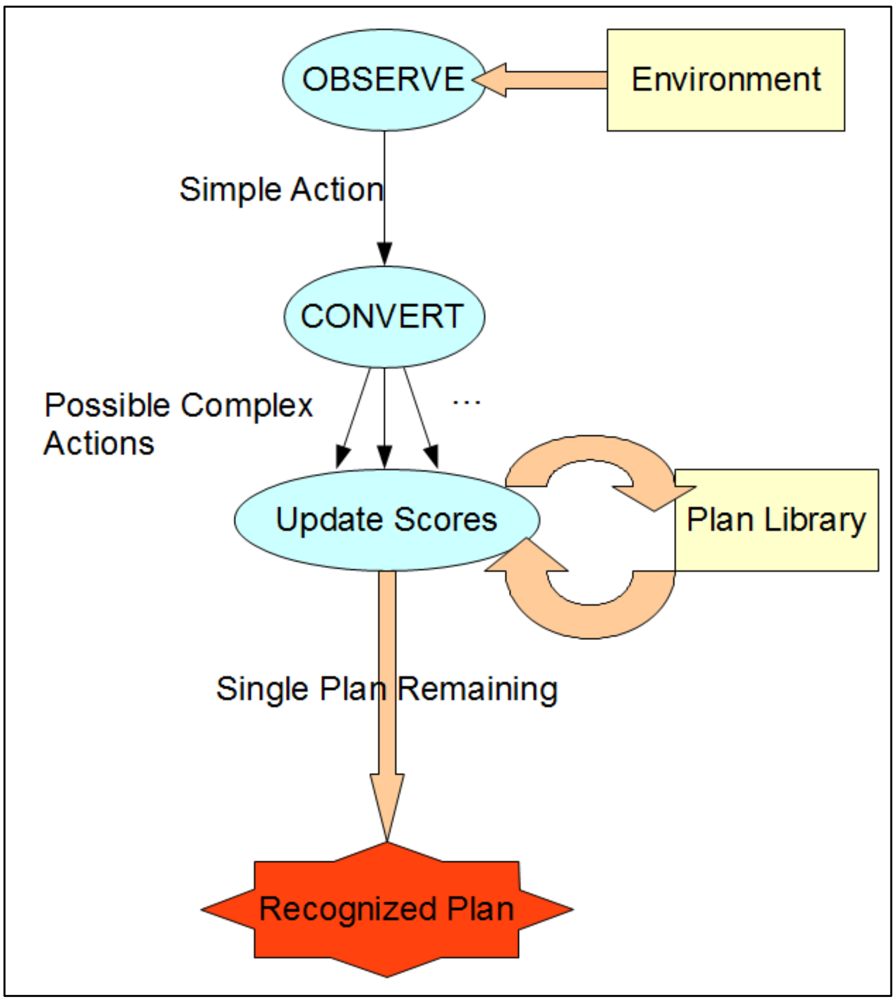


Figure 3.1: Recognition structure overview.

A key component to a multi-agent system is the mechanism which allows agents to interact. In some cases, the agents may be adversaries and pursuing their own individual or team goals. On the other hand, in a cooperative environment, agents may be trying to achieve an increase in the overall utility of the system. The much researched idea of plan recognition can be used in either of these situations. In plan recognition, an agent attempts to determine the plan that another agent is following. A plan is a set of actions which leads to a result. Plan recognition is used in situations where communication between agents is to be avoided. An example of this is a military operation where communications may be intercepted by an enemy agent. An overview of the plan recognition process is shown in Figure 3.2.

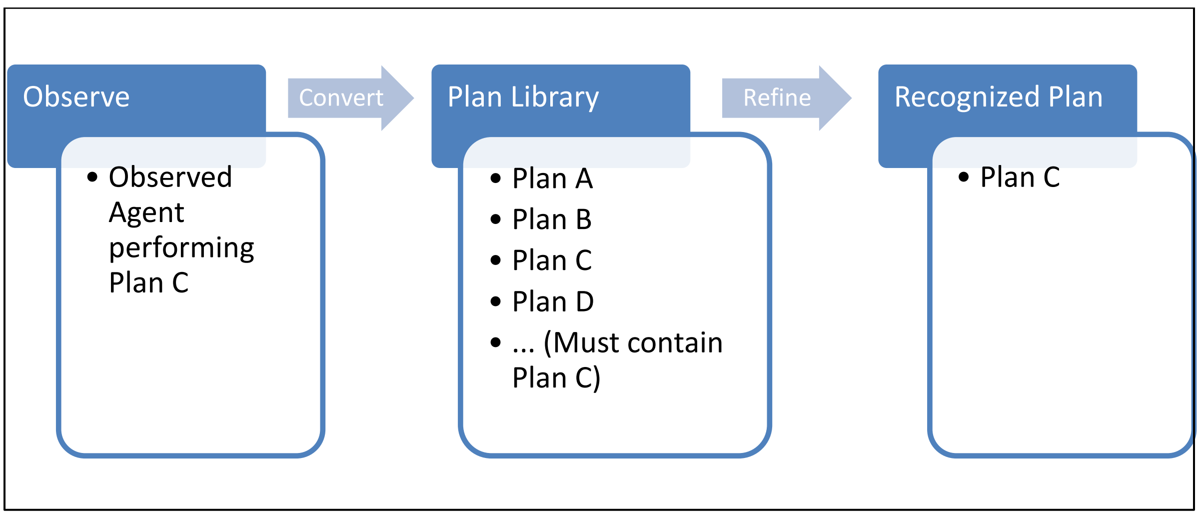


Figure 3.2: The plan recognition process.

Expanding on the idea of plan recognition in a cooperative environment, we propose the utilization of the construct of intent recognition in multi-agent systems. The goal of an agent performing intent recognition is to aid the other agents in the system. The agent does this by attempting to follow a plan that has the same intent, or overall goal, as the other agents in the system. Similar to plan recognition, intent recognition aims to minimize the number of communications between agents due to the cost associated with such communication. An overview of the intent recognition process is shown in Figure 3.3.

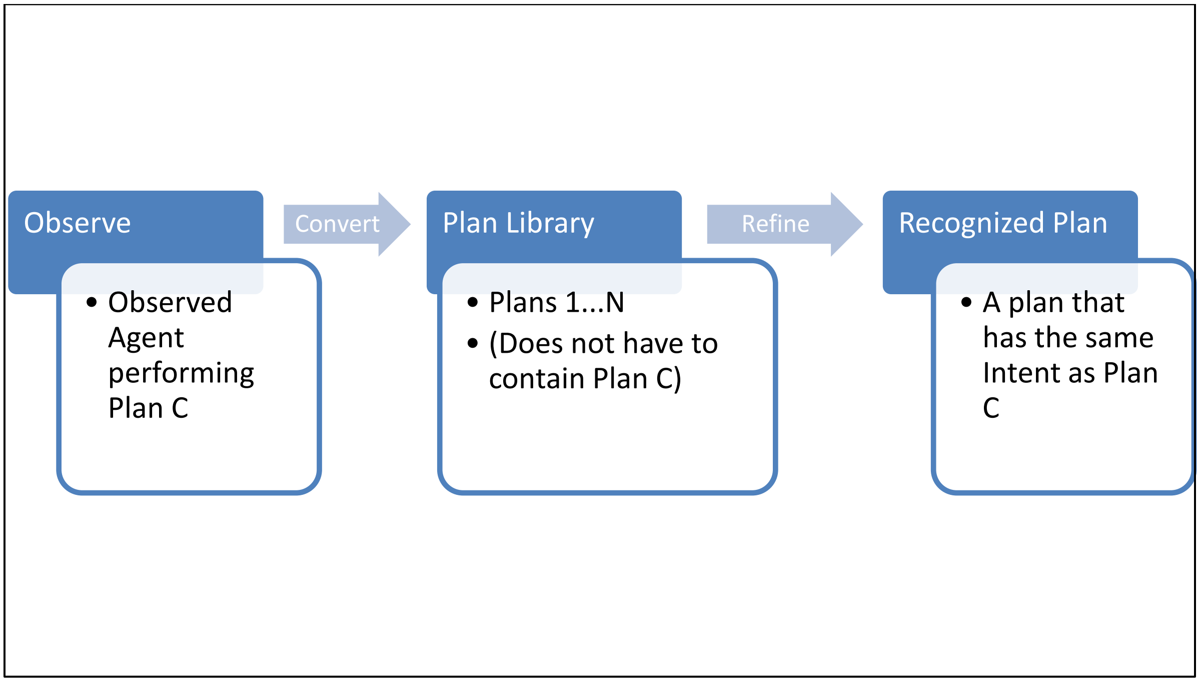


Figure 3.3: The intent recognition process.

Plan recognition attempts to determine the plan, or set of steps, that an observed agent is following. On the other hand, intent recognition attempts to determine the intent, or overall goal, of the observed agent.

We return to the previously mentioned example of a person carrying a heavy bag into an apartment building. The person may have to pause in order to place the bag on the floor, unlock the door with the key, and open the door. The door might need to be propped open with something as the person picks up the bag and walks through the door. Then the person would have to put down the bag down again in order to shut the door. This plan would take less time if the person's acquaintance was inside the apartment building and saw the person approaching through the window. By observing the person, the acquaintance could open the door for the person without explicitly being asked. The person's new plan would be to walk to the door of the apartment building and walk through the door. Because of the assistance of the acquaintance, the overall task time and the number of plan steps were reduced. Another aspect to consider is that there was no communication in this case. Sometimes communication is not possible, such as in the cases where the person carrying the bag did not have a free hand to signal to the acquaintance or the acquaintance could not hear the person through the door. In either case, communication was not possible or necessary for intent recognition to take place.

In contrast to intent recognition, if the acquaintance was using a plan recognition approach, the analysis of the situation would be different. The exact observed plan includes entering the apartment building from the outside. Since the acquaintance is located inside the building, he or she is unable to assist with that task. It is possible that the acquaintance has prior knowledge that certain actions are helpful in various situations. For example, the acquaintance may have known from previous experiences that it is helpful to open the door for someone who is trying to enter the building. However this previous knowledge is not part of the core plan recognition process and, rather, is an addition to it.

3.1 Intent Recognition Structure

Intent recognition takes the idea of plan recognition and expands on it so that agents are better able to cooperate. It builds on the idea that if two agents approach the same task using different techniques, the problem may be solved in a more efficient manner. Even in the case where one agent has different capabilities than another agent, the agents are able to work together to reach a common goal.

Similar to plan recognition, intent recognition also requires agents to keep a plan library. However, it is not a hindrance if all agents in the system have different plan libraries based on their individual capabilities.

The agent performing intent recognition makes observations about another agent in the environment. Instead of trying to find an exact match between the observations and a plan in the plan library, the agent is attempting to find a plan that is similar to the steps that are being observed.

Plans are considered similar to the observations on two criteria. The first consideration is the number of observations that appear in a particular plan. A plan where 75% of the observations occur is less similar than a plan where 90% of the observations occur. When implementing intent recognition, the second plan would have a higher score. The second factor when determining intent is the number of actions in the plan that have not yet been observed. This would also affect the score of the plan when implementing intent recognition.

An example is provided here. In this example there are 10 actions in the system. They are denoted by the numbers 1 through 10. A plan is represented as a list of these actions, for example {1, 2, 3}. In this example, the intent recognition agent's plan library is shown in Table 3.1.

Plan Name	**List of Actions**
PLAN A	{1, 2, 3, 4, 7}
PLAN B	{6, 7, 8}
PLAN C	{1, 4, 7}

Table 3.1: Example of an intent recognition agent's plan library.

The observations made by the intent recognition agent of the observed agent are found in Table 3.2

Timestep	Observation
1	4
2	7
3	9

Table 3.2: Example of observations made by an intent recognition agent.

In this case, plans A and C are more similar to the observation because they both include two out of the three observed actions. The base scores of plan A and C are higher than plan B. Plan C is chosen as the plan with the most similar intent because there is only one action in plan C that has not yet been observed, while there are three actions in plan A that have not been observed. In this case, plan C would have the highest score of the plans in the plan library when performing intent recognition.

Plan recognition would look at the above example in a different manner. The third observation of 9, which was not in any of the plans in the plan library, would be seen as an outlier to all of the given plans and would reduce their overall scores. Intent recognition does not reduce score for outlying observations. After the three observations, plan recognition would not have enough information to recognize a plan. Plans A and C would have the same number of observations corresponding to them. Even if the next observation was 1, and all three steps (1, 4, and 7) from plan C were seen, these three steps are also seen in plan A. Additional reasoning capabilities are needed in order to distinguish these two plans. This is accomplished in the implementations of our intent recognition and plan recognition systems. Our plan recognition system attempts to make further observations in order to distinguish the plans. Our intent recognition system uses factors such as plan length to distinguish between plans.

Chapter 4 Research Methodology: Box Pushing

We designed and built a collective box pushing simulation environment to test the intent recognition concept. We ran experiments in order to compare intent recognition and plan recognition in a multi-agent system (Ahmad and Agah, 2013).

All of the code for experiment series 1 was written in Java using Repast Simphony (Repast, 2012), which is an open source Java-based agent modeling and simulation platform. Repast Simphony was integrated into the Eclipse Platform, which allows for development and debugging of Java projects. Repast Simphony was also used to create the visualizations for the project. A total of 6,677 lines of code were written for the box pushing experiment series.

4.1 Experimental Setup

In order to evaluate intent recognition, a collective box pushing simulated world was implemented. The world consists of an NxN grid. Boxes are represented by squares, agents are represented by circles, and obstacles are represented by triangles. Agents can move in one of the four cardinal directions. Agents are also able to push boxes in one of the four cardinal directions, if they are adjacent to the box and lined up to face the intended direction. The grid does not have wrap around borders so if a box or agent encounters the edge of the grid, it will stay in place until it moves or is moved in another direction.

A 50 by 50 grid was built for each run. Populations of agents on a grid pushed various colors of boxes in one of the four cardinal directions. There were four box colors in all: red, yellow, blue, and green. Once a box reaches the edge of the grid, it is considered to be in the "moved" state and can no longer be pushed. Depending on the dataset, agents are tasked with moving one or more specific box color groups to a given side of the grid. An example of this would be a plan where all yellow boxes are to be moved to the east. There were three types of agent groups (no recognition, plan recognition, and intent recognition) which are explained later. All agents work simultaneously to complete tasks in a cooperative manner. The groups of colored boxes and varying goal direction were used for recognition purposes. In other words, an agent performing recognition had to determine which color of boxes was being moved and also in which direction the boxes were to be moved. An illustration of the grid can be seen in Figure 4.1. The appropriately colored squares represent red, blue, yellow, and green boxes. Circles represent agents, with agents performing intent recognition in purple and observed agents shown as gray. Triangles represent obstacles.

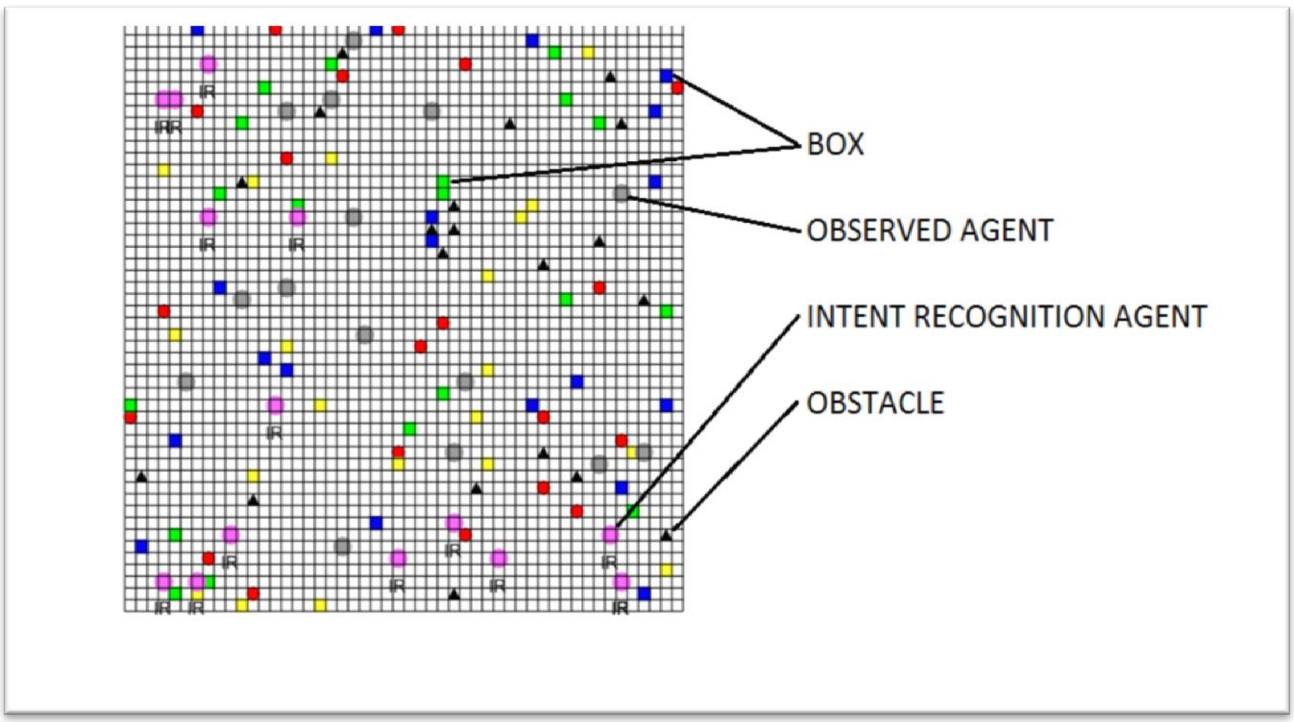


Figure 4.1: Screenshot of the box pushing simulation.

For each data set, three groups of experiments were done. The non-recognition group had only observed agents and was used for comparison; the intent recognition group had observed agents and intent recognition agents; and the plan recognition group had plan recognition agents and observed agents. For each of these groups, the experiments were run with agent populations of 5, 10, 15, 20, 25, and 30. The experiments were run five times for each agent population size. There were 11 data sets, leading to a total of 990 experimental runs. These are listed in Table 4.1.

Parameter	Value	Notes
Groups	3	Non-Recognition, Plan Recognition, Intent Recognition
Agent Population	6	5, 10, 15, 20, 25, 30
Experimental Runs Per Population	5	N/A
Data Sets	11	N/A

Table 4.1: Experiment Series 1 experimental parameters.
Total experiments run: 3x6x5x11 = 990.

As mentioned above, there were 11 different data sets. The obstacle delay, start energy, start plan, boxes per color, and number of obstacles could be changed by data set as listed in Table 4.2. A description of the starting plans is given later in this section.

Data Set	Obstacle Delay	Start Energy	Start Plan	Boxes Per Color	Number of Obstacles
1	5	2000	YBE	20	20
2	5	1000	YBE	30	30
3	5	500	YBE	50	50
4	15	500	YBE	30	90
5	15	500	YBE	25	180
6	5	500	YBE	50	300
7	10	500	YEBN	50	300
8	10	500	YEBN	30	90
9	5	1000	YEBN	30	30
10	5	1000	YEBNRW	30	30
11	5	20000	ALL E	20	20

Table 4.2: Experiment Series 1 Dataset Values.

Values for Obstacle Delay ranged from 5 to 15. If an agent lands on an obstacle, each timestep that they are delayed is multiplied by the cost of what it was doing when it reached the delay. For example, if the Obstacle Delay is a value of 5, an agent that was moving (which is set at a cost of 2 in our experiment) spends 10 units of energy moving across that obstacle. If the agent was pushing a box (which is set at a cost of 3 in our experiment), the agent would spend 15 units of energy while moving across the obstacle. Details on the cost of actions can be found in Table 4.4.

We use Repast Simphony's internal time system to measure the passage of time in the experiments. Every simulation begins at the 0^{th} timestep and the timesteps increment by 1 thereafter. Every agent can perform one action at each timestep, whether it is to observe, move, or perform another action in their plan library.

In our research, agents performing recognition have a set time limit, referred to as TICKS_BEFORE_COMMUNICATION, by which they must complete recognition. The value used for TICKS_BEFORE_COMMUNICATION was varied throughout the data sets so that the effects could be analyzed under different contexts. If the agent has not completed recognition by this time, it then stops the recognition process and communicates with the observed agent.

Start energy ranged from 500 to 20,000. The justification for this is as follows. 500 was used as the minimum and still have agents complete tasks. If the TICKS_BEFORE_COMMUNICATION is set to 200, an agent can observe for the maximum 200 timesteps and still have energy to go across one direction of the 50x50 grid. Since an agent uses one unit of energy for each timestep that it observes, this equals 200 energy units. Moving has a cost of 2 so moving across the 50x50 grid in either the X or Y direction costs 50 (grid dimension) * 2 (moving cost) = 100. Pushing a box has a cost of 3, so pushing a box from one side of the grid to the other side of the grid costs 50 (grid dimension) * 3 (box pushing cost) = 150 units of energy. Adding these values, 200 + 100 + 150 gives 450 energy units needed for observation, traversing the grid once, and pushing a box across the grid once. An extra 50 units were given for obstacles that were encountered for a total of 500 energy units. Despite this, not all agents were able to complete a task in every experimental run due factors such as encountering an excessive number of obstacles or spending energy moving towards a box that is moved by another agent before the first agent reaches it. This minimum value of 500 provided us with a value that was useful in the analysis of our results. The maximum value used for start energy was 20,000. This was used as a "max value" or in other words a case where energy was not a concern and the agents were able to complete their tasks without having energy as a constraint. The exact values for the lower and upper range of this variable were determined via preliminary testing of the system.

Four different starting plans for the observed agents were used across all of the experiments, which can be seen in Table 4.3. For each experimental run, all observed agents follow the same plan.

Plan Abbreviation	Plan Description
YBE	Push all yellow boxes East
YEBN	Push all yellow boxes East and Blue boxes North
YEBNRW	Push all yellow boxes East, Blue boxes North, and Red boxes West
ALL E	Push all available boxes to the East

Table 4.3: Box Pushing: Starting plans of the observed agents.

The value for Boxes Per Color ranges from 20 to 50. There are always 4 colors so that means the number of boxes on the grid ranges from 80 (20 * 4) to 200 (50 * 4). This range was set so that in less complex situations, it is possible for smaller agent populations to complete their tasks, while in more complex situations, even the larger agent populations are unlikely to complete their tasks. The exact values for the lower and upper range of this variable were determined via preliminary testing of the system.

The number of obstacles ranges from 20 to 300. The grid is 50 by 50 so there are 2,500 locations. This means that the percent of locations that contain obstacles range from 0.8% (20 / 2500) to 12% (300 / 2500). At the lower percent, it is likely that an agent can complete its tasks without encountering an obstacle. At the higher percent, most agents encounter at least one obstacle. The exact values for the lower and upper range of this variable were determined via preliminary testing of the system.

The idea of obstacles was introduced into the simulation environment in order to compare the differences between plan recognition and intent recognition. The number of obstacles in the environment varied from dataset to dataset. The starting location was randomly assigned by the system at the beginning of each simulation. A snapshot of the simulation is shown in Figure 4.1.

When an agent encounters an obstacle, it takes a certain number of timesteps to traverse over it. This number was referred to as the Obstacle Delay. The delay was varied by dataset. An agent that attempts to cross an obstacle is hindered in two ways. The most obvious way is the time taken to complete a task. However, it is also a drain on energy. The longer an agent stays in one location, the more energy is used. In the case where an agent is pushing a box while crossing an obstacle, more energy is used by the agent.

4.2 Agents

This section provides further details on the agents utilized in this research. Agents are either not capable of recognition, capable of plan recognition, or capable of intent recognition.

Agents begin each experimental run with a predetermined numerical value that represents the amount of energy that they have throughout the experiment.

4.2.1. Types of Agents

Three types of agents are considered: observed agents, plan recognition agents, and intent recognition agents.

Observed agents are assigned a plan to follow at the beginning of the experimental run. They have no reasoning capabilities. The observed agents continue following the plan until they either run out of energy or the task is completed. For example, an observed agent can be given a plan to push all of the yellow boxes on the grid to the East. By observation, the agent locates a yellow box that is not on the East side of the grid and then attempts to push it in the proper direction. Once the agent has finished, it will locate and push another one. The agent will continue until it either runs out of energy or there are no longer any yellow boxes which are not on the East side of the grid.

Plan recognition agents begin by choosing an observed agent on the grid to observe. The plan recognition agent then makes and stores observations of that agent to determine which plan is being executed. Further details of this process are provided later. Once the plan is determined, the plan recognition agent then begins executing that plan. It continues until all plan objectives have been met or it runs out of energy. If all plan objectives have been met and the agent has energy remaining, it will begin the observation process again.

Intent recognition agents also begin by choosing an observed agent on the grid. The intent recognition agent then makes observations until it is able to recognize a plan with similar intent to the observed agent's intent. Further details of this process are provided later. This plan with similar intent is then executed until all plan objectives have been met or the agent runs out of energy. If all plan objectives have been met and the agent has energy remaining, it will begin the observation process again.

Plan recognition differs from intent recognition in that a plan recognition agent attempts to determine the exact plan that the observed agent is executing. Intent recognition agents attempt to find a plan in their plan library that has the same overall intent as the plan that the observed agent is executing. The plan determined by the intent recognition agent may or may not contain actions that it has not yet observed.

4.2.2 Energy

Agents begin a simulation experiment with an energy level which varies by data set. Various actions use different amounts of energy. When an agent no longer has any energy remaining, it will remain in its current location and will no longer perform any observations or actions.

An agent requires energy to perform actions in the system. In the interest of designing a parsimonious system, costs for actions are constant among all data sets. This allowed more focused analysis on intent recognition instead of system specifics. In future research, these costs can be varied across the data sets. These costs are shown in Table 4.4.

Action	Cost
Observe	1
Move	2
Push Box	3
Communicate	10
Destroy (Remove) Obstacle	3

Table 4.4: Experiment Series 1 Box Pushing - Action costs.

Costs for communication are kept relatively high to simulate an environment where it is to be avoided, such as when an adversarial team may intercept communications.

4.3 Recognition

The agents in the system that were able to perform recognition either performed intent recognition or plan recognition. In this way, a comparison can be made between the two recognition types.

4.3.1 Plan Recognition

A parsimonious plan recognition system was designed and implemented for this study. At the beginning of an experimental run, each plan recognition agent randomly picks an observed agent to monitor.

Every plan recognition agent in the system has access to a copy of the same plan library. Each plan is assigned a score of zero at the beginning of the simulation. When an observation is made by a plan recognition agent that coincides with a given plan, the plan score is increased for that agent's copy of the plan. If the observation does not coincide with the plan, the plan score is decreased.

A single observation can increase the score of multiple plans. For instance, an observation is made that the observed agent moved north. In this example, the plan recognition agent's plan library consists of three plans, which can be seen in Table 4.5.

Plan Name	Set of Plan Steps
GREEN EAST	{MOVE TO A GREEN BOX, PUSH THE GREEN BOX EAST}
GREEN WEST	{MOVE TO A GREEN BOX, PUSH THE GREEN BOX WEST}
BLUE EAST	{MOVE TO A BLUE BOX, PUSH THE BLUE BOX EAST}

Table 4.5: Example of a plan library for a box pushing plan recognition agent.

The plan recognition agent then observes the environment. If there is a green box to the north, the scores for GREEN EAST and GREEN WEST will be increased. If not, the scores for these two plans will be decreased. If there is a blue box to the north, the score for BLUE EAST will be increased, otherwise it will be decreased. If there is at least one green box and one blue box to the north, all of the plans will have their score increased. Similarly, if there are no green or blue boxes to the north, all of the plans will have their scores decreased.

Once a single plan has a higher score than any of the other plans, this is determined to be the "best" possible plan. If the agent is not able to recognize a plan within a given time frame, the plan recognition agent then communicates with the observed agent.

Whether by plan recognition or communication, the plan recognition agent begins to follow the recognized plan in order to aid the observed agent.

4.3.2 Intent Recognition

Intent recognition differs from plan recognition in that the agent attempts to determine the observed agent's intent instead of determining the plan it is following.

Intent recognition begins similarly to plan recognition. Observations are used to update scores in the intent agent's plan library. In addition to this information, the number of steps that have been observed of a particular plan is recorded. If an action is observed that is not in the plan, the plan score is not reduced.

To calculate the score of a plan, the intent agent stores a value called the original score. This is increased every time an action is observed that pertains to the plan. The original score is then combined with the number of plan steps that have been seen to calculate the *adjusted score*.

$$\text{Adjusted Score} = O + I/S$$

When (S1 == S2) the value of S is 1, otherwise S is given the value of S1 – S2 where S1 is the number of steps in the plan and S2 is the number of steps in the plan that have been observed. O is the original score and I is an adjustable intent recognition bonus. When S1 and S2 are equal, the adjusted score is O + I. Future research can change this value to see how this would affect intent recognition.

When there is a single plan in the plan library that has a higher adjusted score than any of the other plans, this plan is selected as the most likely to have the same intent as the plan that the observed agent is following.

Similar to plan recognition, if a plan is not recognized before the set timeframe, the intent recognition agent communicates with the observed agent in order to determine its plan. If the intent recognition agent has communicated, it will attempt to aid the observed agent by following the same plan that the observed agent sent in the reply to its query. Intent recognition and plan recognition behave in the same manner if communication is necessary. However, if intent recognition was completed prior to the communication, the intent recognition agent could execute a different plan than the observed agent in order to aid it.

An example of this would be if an intent recognition agent is observing an observed agent which is pushing yellow boxes to the east. The intent recognition agent may decide that the plan it its library with the closest intent is the one which first removes all the obstacles between the yellow boxes and the eastern wall and then proceeds in pushing the boxes. A plan recognition agent has the same plan library, which also includes obstacle removal. However, since the observed agents are incapable of obstacle removal, the plan recognition agents will never observe this action and therefore will never recognize a plan that includes it.

4.4 Actions, Plans and the Plan Library

An agent draws its knowledge about how to interact with the environment from its plan library. A plan library is a collection of one or more plans, where a plan is a collection of one or more actions.

4.4.1 Actions

Actions in the system were translated into numerical values. Similar to the research from Honeng and Nevatia (2001), we assume that there are two types of events. In our research, these are referred to as simple events and complex events. An agent that is performing plan recognition or intent recognition makes observations in simple events. There are 10 simple events used in the box pushing simulation: PUSH_BOX_WEST, PUSH_BOX_EAST, PUSH_BOX_NORTH, PUSH_BOX_SOUTH, MOVE_EAST, MOVE_WEST, MOVE_SOUTH, MOVE_NORTH, DESTROY_OBSTACLE, and NONE.

In terms of complex events, these are handled as follows. Based on observations about the environment, the agent then converts these observations into complex actions. For example, an agent is observed to be performing the MOVE_NORTH simple action. If there are red boxes to the north of the agent, this would be interpreted as a MOVE_TOWARDS_RED_BOX complex action. If there were also one or more blue boxes to the north this would also be interpreted as MOVE_TOWARDS_BLUE _BOX. In this way, one observation of a simple event can be translated into one or more complex events.

4.4.2 Plans

As the literature indicates, plans are collections of complex actions. For example, the plan called “PushRedBoxesNorthPlan” is made up of the complex actions MOVE_TOWARDS_RED_BOX and PUSH_RED_BOX_NORTH.

Another example is the plan called “PushRedBoxesEastPlanDF” which consists of MOVE_TOWARDS_RED_BOX and PUSH_RED_BOX_EAST, and DESTROY_DELAY_AHEAD. The DESTROY_DELAY_AHEAD action determines if there are any obstacles between the boxes and the target location, which in this case is the north side of the environment. The agent then proceeds to destroy those obstacles.

4.4.3 Plan Library

A plan library is a collection of plans. These are the plans that the agents are aware of and are capable of executing. For each dataset, the observed agents are assigned a single plan in their plan library and they execute this plan until the overall task is complete, or until they run out of energy. Both intent recognition and plan recognition require plan libraries. One advantage of intent recognition over plan recognition is that the intent recognition agent’s plan library does not have to contain the plan that is being observed in order for the agent to recognize a plan. Another advantage of intent recognition is that even if the observed plan is in the plan library, the intent recognition agent is capable of selecting a different plan from its library if the agent determines that both plans have the same intent.

For each dataset, the agents performing intent recognition and plan recognition were given the same plan library. Both intent recognition agents and plan recognition agents have plans that contain information about destroying obstacles in the environment. However, because the observed agents do not have the capability to destroy obstacles, neither the intent recognition agents nor the plan recognition agents will ever make an observation along these lines. However, an intent agent may still chose to perform a plan that includes destroying obstacles if the plan is seen as having the same intent as the observed plan. Even though intent recognition agents do not observe destroying obstacles, they have the capacity to so and may implement this action. In systems where agents are performing plan recognition, this would not happen. In the plan recognition systems, plan recognition agents would not destroy obstacles because they would not observe this action. In this way, intent recognition has an advantage over plan recognition.

The “destroy obstacle” action was a deliberate addition. We anticipate that the intent recognition agents will recognize plans with the "destroy obstacle" action and the plan recognition agents will not. Our justification for this is that by the definition of plan recognition, no matter what the plan library, plan recognition agents will only select plans that only contain actions that they have already observed. So, although plan recognition agents have the same plan library as intent recognition agents, if they do not observe obstacle destruction, then they will not select the plans which include destroying obstacles. If they observe destruction, they will select plans which include the destroy obstacle action. Our observed agents cannot perform obstacle destruction. Intent recognition agents, on the other hand, are able to use unobserved actions in their recognized plan which is why they sometimes, but not always, recognize plans with obstacle destruction. Therefore, the extraneous action of "destroy obstacle" was purposefully added to the plan library of both plan recognition and intent recognition agents to see whether it would ever be recognized or not. As shown later, the results indicate that the plan recognition agents did not select plans with obstacle destruction and that intent recognition agents sometimes selected plans with obstacle destruction.

4.5 Variables

Two dependent variables and 10 independent variables were considered in this research. Some independent parameters were varied by dataset while others were not. These are explained below.

4.5.1 Dependent Variables

The primary dependent variables considered in this research are simulation time and percent completed.

1. *Simulation Time* – The number of timesteps that pass between the beginning of the simulation and the last time that a box was correctly moved.
2. *Percent Completed* – The cumulative percent of boxes correctly moved up to and including the simulation time. In order to have a 100% completion rate, all the boxes that the observed agents intend to move must be moved to their correct location.

An example scenario is presented to describe the term "percent completed," where the observed agents are executing a plan where all yellow boxes must be pushed to the east side of the environment. There are five yellow boxes and five blue boxes on the grid. The goal is to move all five of the yellow boxes to the east.

At timestep two, a yellow box is moved to the east.
At timestep five, another yellow box is moved to the east.
At timestep six, a yellow box is moved to the north.
At timestep seven, a yellow box is moved to the east.

In this example, three yellow boxes were moved correctly to the east side. One yellow box was moved incorrectly to the north side. One yellow box was not moved. This means that three out of the five boxes were moved correctly giving a completion percentage of 3/5 or 60%. Since the starting plan did not include blue boxes, they were not considered in the calculation.

4.5.2 Independent Variables

In total, we have 10 independent variables in our first set of experiments. These fall in two categories. In the first category, variables are varied depending on the data set. In the second category, independent variables do not depend on the data set.

In this research, five variables change depending on the data set:

1. Obstacle Delay- The amount of time it takes an agent to traverse an obstacle. When an agent encounters a non-destroyed obstacle without the specific intent to destroy it, this is the number of timesteps it spends crossing the obstacle.

2. Start Energy – The amount of energy an agent has at the beginning of the simulation.

3. Start Plan- The plan that the observed agents are following during the simulation.

4. Boxes Per Color – The number of boxes per color. There were always four colors so the total number of boxes is Boxes Per Color multiplied by 4.

5. Num Obstacles – The number of obstacles at the beginning of the simulation.

In addition, there are five other independent variables:

1. *Rec Type* – The recognition type that a particular agent is performing. The choices are plan recognition, intent recognition, or none. A recognition type of none denotes an observed agent.

2. *When Determined* – The time at which a particular agent completed either plan or intent recognition.

3. *Num Agents* – The number of agents in each agent group of a simulation. For example, if the number of agents is 5 in a run with plan recognition agents, there would be 5 plan recognition agents and 5 observed agents.

4. *Communicated* - Whether a particular agent communicated or not during a simulation.

5. *Plan Determined* – Whether a particular agent was able to determine a plan during a

simulation. A plan can be determined by plan recognition, intent recognition, or through communication.

4.6 Complex Scenarios

In cases where recognition can easily be done, for example in simple scenarios, it would be difficult to examine the differences between intent recognition and plan recognition. To keep the analysis comparable, all intent recognition and plan recognition agents have the same plan libraries. These libraries include plans that are identical except for the addition of the DESTROY_OBSTACLE complex action. Observed agents in this research cannot perform this action, whereas intent recognition and plan recognition agents can. This makes the recognition process for plan recognition agents complex, as illustrated in the following example:

A plan recognition agent has a library which can be seen in Table 4.6. It selects an observed agent on which to perform recognition. The observations are shown in Table 4.7.

Plan Number	Set of Plan Steps
P0	{MOVE_TO_YELLOW_BOX, PUSH_YELLOW_BOX_EAST}
P1	{DESTROY_OBSTACLE, MOVE_TO_YELLOW_BOX, PUSH_YELLOW_BOX_EAST}

Table 4.6: Sample plan library for a plan recognition agent in the box pushing example.

Timestep	Observation
T0	Observed agent moves towards yellow box
T1	Observed agent moves towards yellow box
T2	Observed agent pushes yellow box east
T3	Observed agent pushes yellow box east
T4	Observed agent pushes yellow box east

Table 4.7: Sample observations made by a plan recognition agent in the box pushing example.

Since the order of actions does not matter in our system, the plan recognition agent does not know whether the observed agent is simply pushing the yellow boxes east and following P0 or whether the agent is following plan P1. The plan recognition agent is unable to determine whether P1 is being followed and there are no longer any removable obstacles on the grid, or if the observed agent is following the plan steps out of order. In reality, observed agents do not have that capability and the plan recognition agent is wasting valuable time and resources.

An intent recognition agent, on the other hand, may select P0 after the first one or two observations. The agent may also select P1 after the first three or four observations. The difference can be affected by factors including information the agent has collected previously, any observations about the state of the world, and any observations about other agents.

Chapter 5 Experimental Result: Box Pushing

In series 1 of the experiments, 990 experimental runs were completed using two dependent variables (percent completion and time to completion) and 10 independent variables, as described earlier. The box pushing results are presented in this chapter.

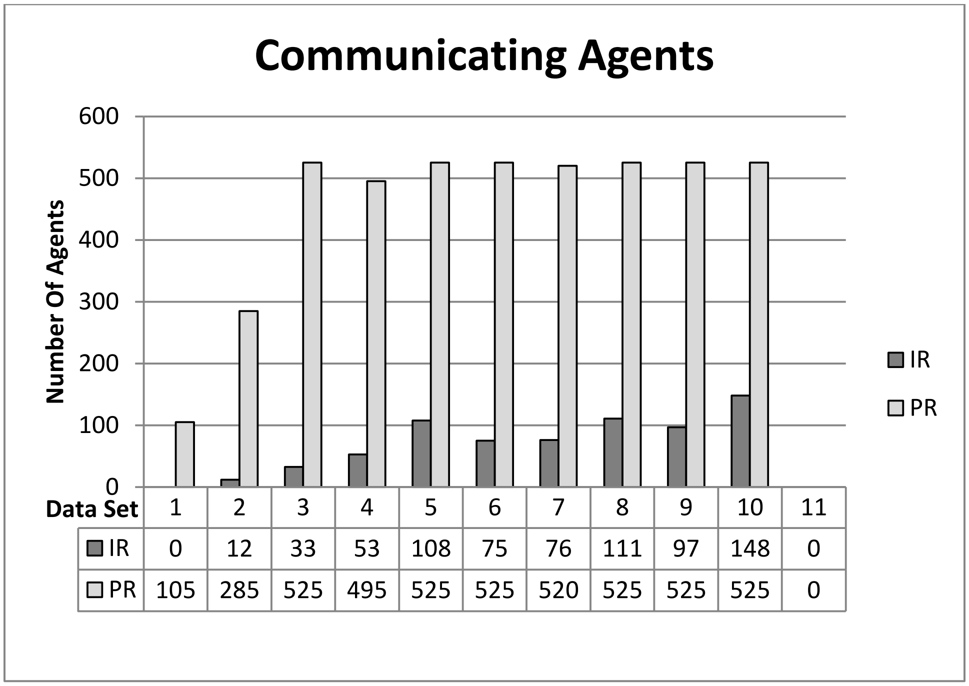
5.1 Communication

The results, shown in Figure 5.1, indicate that intent recognition, as hypothesized in H4, is able to communicate fewer times than plan recognition in a multi-agent environment where exact plans cannot easily be determined. The figure shows that the number of intent recognition agents (IR) and the number of plan recognition agents (PR) which communicated for each dataset. The “no recognition” group was not included because the observed agents do not initiate communication. There were a total of 5,775 agents per recognition type. 713 intent recognition agents communicated. 4,555 plan recognition agents communicated.

The agents performing plan recognition tried to make an exact match between their observations and their plan library. Because of this, often times the plan recognition agents were not able to complete the recognition process before reaching the set time limit.

Agents performing intent recognition were not looking for an exact match between observations and their plan library. Not only did the intent recognition agents communicate a fewer number of times, but a fewer number of intent agents communicated when compared to the number of plan recognition agents that communicated.

There was no communication in dataset 11 because the value of the variable which limits the number of timesteps allowed for recognition before communication takes place, was set to a max value. Therefore, both plan recognition agents and intent recognition agents had unlimited time to perform recognition and did not communicate.

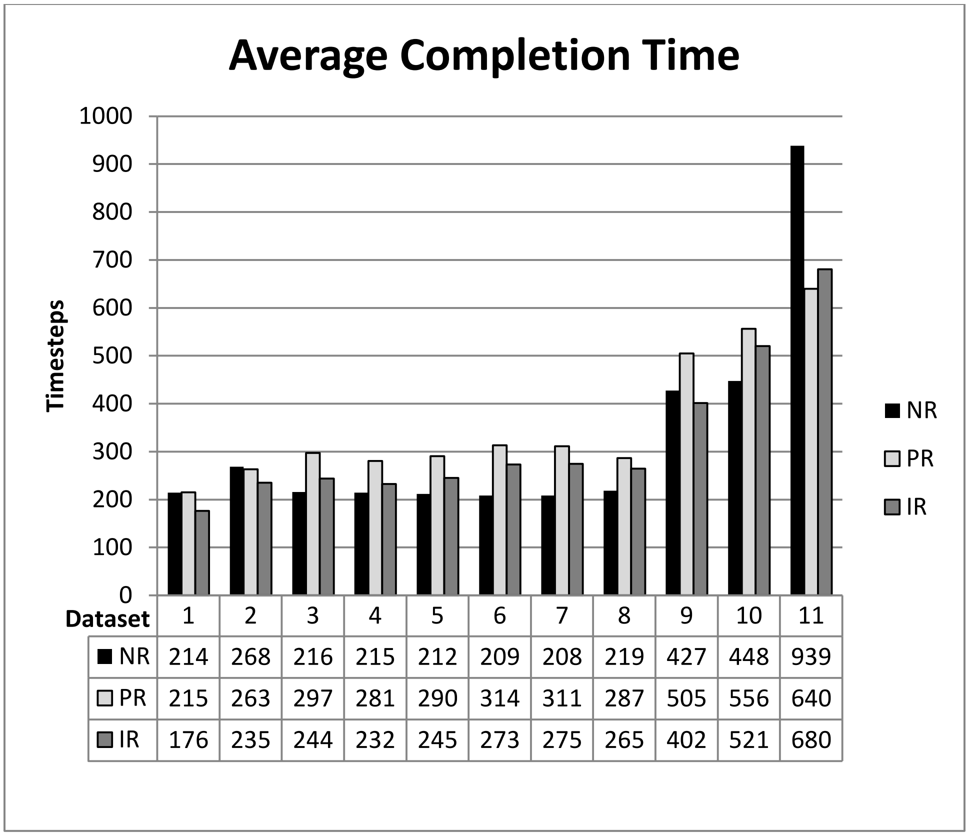


Data Set	1	2	3	4	5	6	7	8	9	10	11
IR	0	12	33	53	108	75	76	111	97	148	0
PR	105	285	525	495	525	525	520	525	525	525	0

Figure 5.1: Communicating Agents: IR (intent recognition), PR (plan recognition).

5.2 Time

We tested hypothesis H1.1. The simulation time was recorded as the last time a box was pushed correctly. Simulations where there was no recognition taking place sometimes had a faster completion time than the simulations where there was recognition of some sort. This is because in the simulations with no recognition, all agents in the environment began their assigned tasks as soon as the simulation began. In the cases where there were agents performing recognition, these agents would first observe for a period of time before beginning to execute their tasks. The results are shown in Figure 5.2. Except for one dataset, intent recognition has a smaller completion than plan recognition.



Dataset	1	2	3	4	5	6	7	8	9	10	11
NR	214	268	216	215	212	209	208	219	427	448	939
PR	215	263	297	281	290	314	311	287	505	556	640
IR	176	235	244	232	245	273	275	265	402	521	680

Figure 5.2: Average task completion time by dataset and recognition type: NR (no recognition), PR (plan recognition), IR (intent recognition). Recogntion type is statistically significant in datasets 1-9.

A regression analysis was performed using the simulation time data. The details of the regression are shown in the Appendix. The recognition type and the number of agents were used as independent variables. The type of recognition was represented using a variable with 0 to represent no recognition, 1 to represent plan recognition, and 2 to represent intent recognition. The dependent variable was completion time.

The results of the regression (the related table can be found in the Appendix) indicate that there are cases where completion time tends to be higher between groups with recognition and groups without recognition, such as can be seen in datasets 6, 7, 8, and 10. This is because the recognition agents spend the beginning of the simulation observing and do not begin working until they have either determined a plan or communicated. Because of this, recognition agents are usually still working after all of the non-recognition agents have run out of energy. The goal is to see whether plan recognition and intent recognition are significantly different and which one is correlated with a higher simulation time.

According to our research, in datasets 1 through 9, recognition type has a significant negative impact on completion time. This means that intent recognition, with the higher variable value (2 = intent recognition), is associated with a lower completion time for tasks. Hence, H1.1 is partially confirmed. More details of this time regression are included in the Appendix.

5.3 Percent Completion

Percent completion was measured as the number of boxes that were moved to the correct location out of the total number of boxes that were to be moved. The results are shown in Figure 5.3.

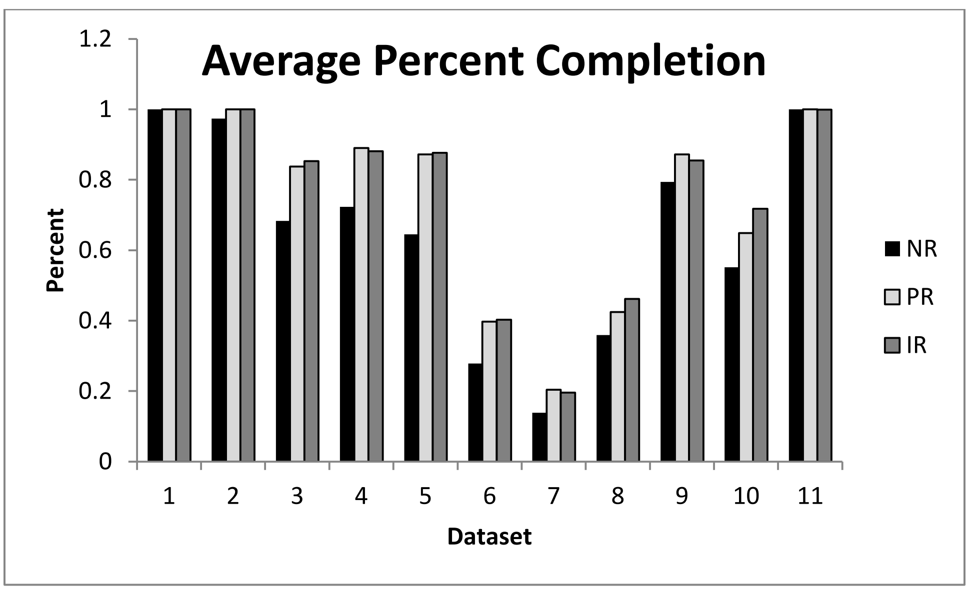


Figure 5.3: Average percent completion by dataset and recognition type: NR (no recognition), PR (plan recognition), IR (intent recognition).

The independent variables used were agent population size and recognition type. Recognition type was a categorical variable with the following values: 0: no recognition, 1: plan recognition, and 2: intent recognition. Datasets 1 and 2 had no variation of percent completion. In all datasets aside from 11, there was a significant t-value for the recognition type. This indicates that recognition type is a significant variable that impacts percent completion. The details of the regression are included in the Appendix.

Regression was then done with only the data for the intent recognition and plan recognition agents. We found that percent completion was not statistically different between plan recognition and intent recognition in most datasets. Our research shows that the plan recognition group and the intent recognition group had a statistically higher completion percentage than the no recognition group.

In dataset 11, intent recognition agents and plan recognition agents had a similar percent completion. Both intent recognition and plan recognition agents had a higher percent completion than non-recognition agents. However, even though intent recognition agents had a comparable percent completion to plan recognition agents, they communicated less and had a faster completion time.

5.4 Accuracy of Recognition

In this research, the results indicate that intent recognition agents performed actions that they had not observed. Intent recognition agents and plan recognition agents had additional capabilities when compared to the agents that they were observing. Unlike the observed agents, these agents were able to identify and destroy obstacles in the environment.

Since the observed agents did not have the capability to destroy obstacles, the plan recognition agents did not have any observations along these lines. For this reason, the plan recognition agents never recognized a plan with the DESTROY_OBSTACLE action in it as the correct plan.

Plan recognition and intent recognition agents had the same plan libraries. The intent recognition agents look for a plan which has an intent close to what they are observing. Sometimes this includes destroying obstacles, sometimes this does not. The plan recognition agents look for a plan that is the same plan as that they are observing. Since the observed agents do not have the capability of destroying obstacles, this is not an observation that could be made by either intent recognition or plan recognition agents. The difference is, for intent recognition agents they sometimes execute steps that they have not directly observed.

Many intent recognition agents chose to follow plans that involved destroying obstacles. Table 5.1 shows the number of obstacles that were removed by intent recognition agents over the 11 datasets.

Dataset	1	2	3	4	5	6	7	8	9	10	11
Obstacles Removed	84	205	437	536	637	1659	1684	554	313	341	0

Table 5.1: Obstacles removed by intent recognition agents by dataset.

Intent agents also dynamically chose other actions that they had not observed. Some of these may have been beneficial depending on the overall system goal while others were not. One example is where there were five red boxes and five blue boxes on the grid. In this example, there is one observed agent whose task it is to move red boxes north. After the intent agent observes a box being pushed to the north, it may decide that all boxes should be moved to the north. If the observed agent's next task is to push all the boxes to the north, this will help to decrease the task time. However, if the observed agent is actually trying to push the red boxes north and the blue boxes east, this will introduce task error into the system.

As Table 0.6 in the Appendix shows, there were 5,418 intent recognition agents that were able to recognize a plan. Out of these agents, 5,403 of them recognized a plan that had the possibility of assisting the observed agents. 710 of the agents recognized a plan that had the possibility of introducing error into the system. This does not indicate the percentage of the agents that were able to recognize a plan because an agent restarts the recognition process once it has completed the cycle of recognition and assisting.

Percent completion was comparable between the intent recognition agents and plan recognition agents while the intent recognition agents had faster task completion times and fewer communicating agents.

Chapter 6 Research Methodology: Cow Herding

Teams of 20 agents faced off in the cow herding domain (Ahmad and Agah, in review). Rules, field configuration, and team size followed the guidelines of the Multi-Agent Programming Contest or MAPC (Dix *et al.*, 2013).

Similar to the box pushing experiment series, the code for experiment series 2 was written in Java using Repast Simphony (Repast, 2012), which is an open source Java-based agent modeling and simulation platform. Repast Simphony was integrated into the Eclipse Platform, which allows for development and debugging of Java projects. Repast Simphony was also used to create the visualizations for the project. Also, Repast internal time measure was used to measure timesteps. A total of 4,274 lines of code were written for the cow herding experiment series.

There were three types of agents in the experiment: non-recognition (or observed) agents, plan recognition agents, and intent recognition agents.

There were six datasets. In every dataset, all of the agents had the same plan library which consisted of 8 plans. In datasets 1 and 2, the observed agents were following plan 2. Plan 2 was considered to be an optimal plan (explanation provided later). In datasets 3 and 4, observed agents were following plan 4, which was considered to be an ineffective plan (explanation provided later). In datasets 5 and 6, observed agents were following a randomly chosen plan from their plan library. Intent and plan recognition agents also had this same plan library that they could select plans from. Table 6.3 shows which observed plans the observed agents were following in each dataset.

Recognition agents were given a time allotment in which to recognize a plan. If the agent was not able to recognize a plan in this time limit, the agent then initiated communication. The goal of intent recognition is to reduce the amount of communication in the system. The time allotment for datasets 1, 3, and 5 was 200 timesteps. The time allotment for datasets 2, 4, and 6 was 50 timesteps.

For each dataset, competitions were run with intent recognition versus plan recognition, intent recognition versus no recognition, and plan recognition versus no recognition. There were also competitions with no recognition versus no recognition to make sure that there was no bias based on which side of the field a team's corral was located. Corrals were located on the north and south sides of the field.

For each of the teams with recognition, the number of recognition agents per team was varied. Trials were run with 5, 10, and 15 recognition agents per team, with the rest of the 20 agent team made up of non-recognition agents. Each configuration was run 6 times.

6.1 Agents

As mentioned earlier, there were three types of agents in the cow herding experiment. Non-recognition (or observed) agents do not have any recognition abilities. They are either assigned a plan to follow or they are able to randomly select a plan from their plan library.

Plan recognition agents perform plan recognition on non-recognition agents on their team. Intent recognition agents perform intent recognition on non-recognition agents on their team. Recognition agents begin by performing recognition on their chosen teammate. Once a plan has been selected by either recognition or communication, the agent then performs the plan.

Once the plan has been executed, the cycle begins again as the agent selects a new agent on which to perform recognition.

6.2 Cows

There were 114 cows on the field at one time. This number was the same as the number of cows in the MAPC example for the cow herding scenario. The cows followed the algorithm given by MAPC as discussed previously. In this algorithm, locations are given weights based on their content. The weights used in this experiment are the same as the weights used by the cows in the MAPC. These are shown in Table 6.1.

Cell Content	**Weight**
Cow	10
Empty	3
Corral	3
Agent	-200
Obstacle	-4

Table 6.1: Weights used by cows.

6.3 Complex Actions

There were seven possible complex actions:

1. "PUSH": If an agent's location is adjacent to a cow's location, the agent is able to push the cow to a neighboring empty cell. The agent moves to the location where the cow was previously located.
2. "HERD_TOGETHER": The agent moves a particular cow towards another cow. This can be by pushing or following. Following a cow moves it in the opposite direction from the agent since the cow algorithm assigns a large negative weight to agents.

3. “HERD_TOWARDS_GOAL”: The agent moves a particular cow towards the agent’s team corral. This can be by pushing or following. Following a cow moves it in the opposite direction from the agent since the cow algorithm assigns a large negative weight to agents.
4. “FLANK”: The agent moves around a group of one or more cows, staying a specified distance away.
5. “HEAD_TOWARDS_COW”: The agent moves towards a particular cow.
6. “BLOCK”: The agent locates itself directly outside the entrance to the goal to act as a barrier.
7. “HEAD_TOWARDS_GOAL”: The agent moves towards its team corral.

6.4 Plan Library

All agents, regardless of recognition type and ability, have the same eight plans in their plan library. Each plan was made up of multiple complex actions. The plans can be seen in Table 6.2.

Plan Name	Plan Steps
Plan1	{HEAD_TOWARDS_COW, HERD_TOWARDS_GOAL}
Plan2	{HEAD_TOWARDS_COW, HERD_TOWARDS_GOAL, PUSH}
Plan3	{HEAD_TOWARDS_COW, HERD_TOGETHER}
Plan4	{HEAD_TOWARDS_COW, HERD_TOGETHER, FLANK}
Plan5	{HEAD_TOWARDS_COW, HERD_TOWARDS_GOAL, PUSH, BLOCK}
Plan6	{HEAD_TOWARDS_COW, HERD_TOWARDS_GOAL, BLOCK}
Plan7	{HEAD_TOWARDS_COW, PUSH}
Plan8	{HEAD_TOWARDS_GOAL, BLOCK}

Table 6.2: Plan library for agents in the Cow Herding scenario.

6.5 Field

The makeup of the field is the same as the makeup of the field of the MAPC example for the cow herding scenario. The environment is a 70 by 70 grid. Each corral is 15 units vertically and 20 units horizontally. One corral is centered along the north wall of the grid and the other is centered along the south wall. A screen shot from the scenario can be seen in Figure 6.1. The green and blue circles represent the teams of agents and the gray circles represent cows. Pink squares denote fences which block of the corrals.

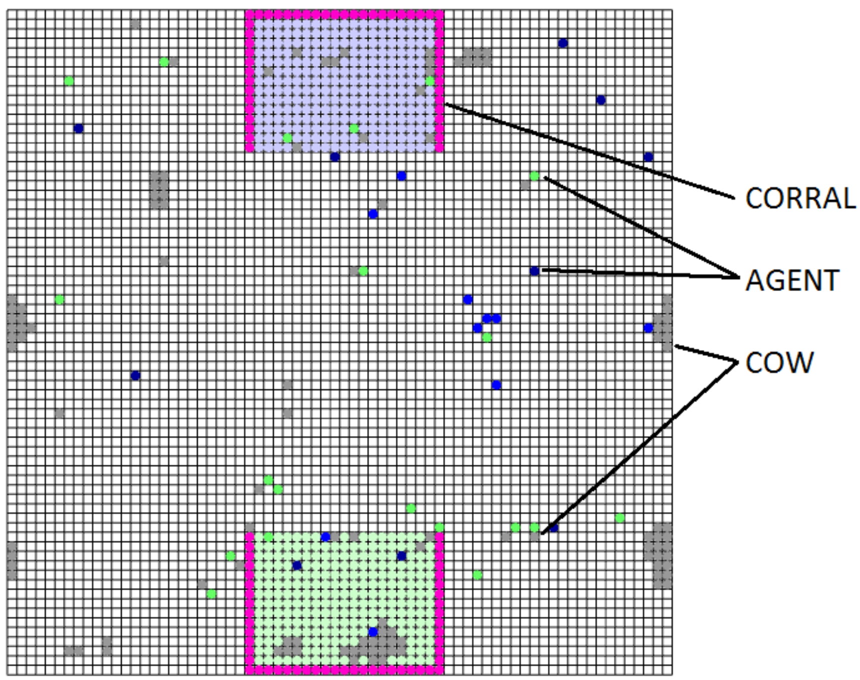


Figure 6.1: Screen capture of Experiment Series 2: cow herding.

6.6 Team Organization

There were three types of teams. Every team consisted of 20 agents, which is consistent with the MAPC rules. The first type of team consisted solely of non-recognition agents. The second type of team consisted of plan recognition and non-recognition agents. The third type of team consisted of intent recognition and non-recognition agents.

For teams with recognition agents, each experiment was run with varied ratios of the types of included agents. Each experiment was run with 5 recognition agents and 15 non-recognition agents; 10 recognition agents and 10 non-recognition agents; and 15 recognition agents and 5 non-recognition agents.

6.7 Time To Communication

Recognition agents are given a set number of timesteps to complete recognition. If the agent is not able to complete recognition in this time, it then communicates with the observed agent. The two time limits used for analysis were 200 timesteps and 50 timesteps. Time to communication was varied in order to test whether altering this variable would have an impact on recognition.

6.8 Datasets

There were six datasets. One factor that differed by dataset was the plan that the observed agents were following. In the first two datasets, these agents followed Plan 2. This plan was used as an example of an "excellent" plan. This plan is a simplified version of the plan by Heßler *et al.* (2010), which won in the MAPC. In the next two datasets, these agents followed Plan 4. This plan was used as an example of a "poor" plan. In this plan, agents attempt to herd cows into large groups, but do not herd them towards a corral. For the final two datasets, the observed agents randomly selected a plan from the plan library.

The other factor that varied by dataset was the time to communication, as discussed previously. In datasets 1, 3, and 5, the time to communication was set to 200. In datasets 2, 4, and 6, the time to communication was set to 50. The details can be seen in Table 6.3.

Dataset	Observed Agent Plan	Time To Communication
1	Plan 2	200
2	Plan 2	50
3	Plan 4	200
4	Plan 4	50
5	Random	200
6	Random	50

Table 6.3: Experiment Series 2 Dataset Configuration.

Every experimental configuration was run six times. In each configuration, the dataset variables, the organization of the teams, and the time to communication were varied.

Chapter 7 Experimental Results: Cow Herding

Our second set of experiments took place in the cow herding domain. We collected data regarding recognition and communication. Efficiency was measured by the number of team wins and the overall score that a team achieved. In this chapter we present an overview of the results, explore how recognition affects the results, discuss how communication affects the results, and describe how adding an agent to a team affects the results.

7.1 Overview of Results

We analyzed the total wins of the three types of teams: teams with plan recognition agents, teams with intent recognition agents, and teams without recognition agents. We studied the total wins, the overall score, and the differences between the team types.

7.1.1 Total Wins

The following figures (Figure 7.1, Figure 7.2, and Figure 7.3) show the number of times that a given team won its match. Teams with intent recognition agents won more matches than teams with plan recognition, each time they competed.

When teams with plan recognition agents competed against teams that did not have any recognition agents, the teams with plan recognition agents lost more matches in every dataset except dataset 3 (Figure 7.2). We will explore the aspects of dataset 3 that were favorable for plan recognition agents, which are “plan followed” and “time to communication.”

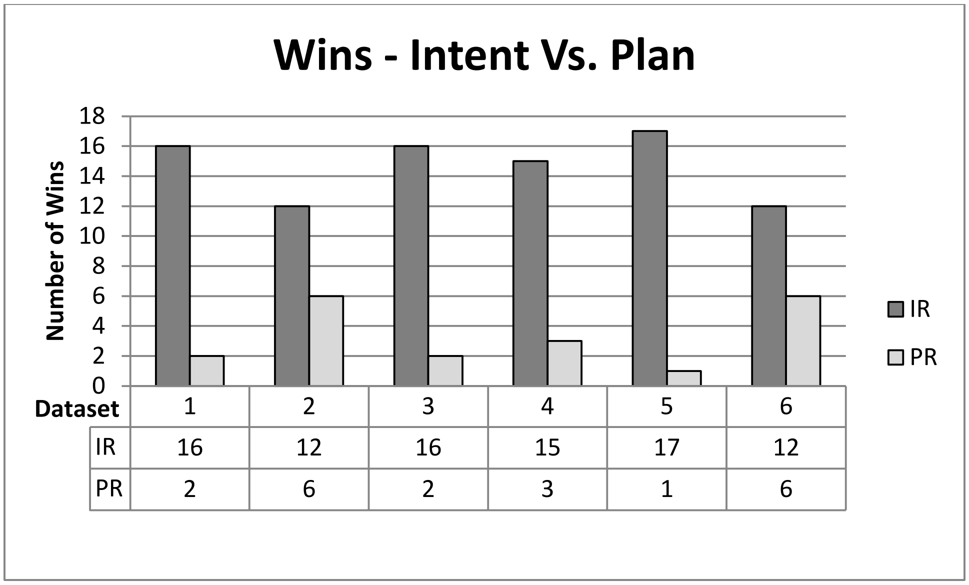


Figure 7.1: Number of wins for intent recognition teams versus plan recognition teams.

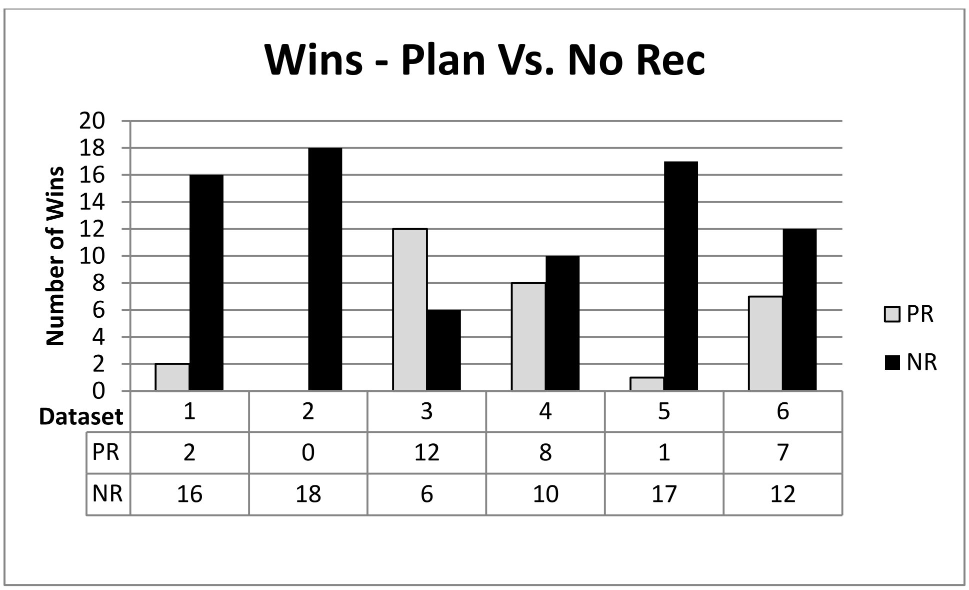


Figure 7.2: The number of wins for plan recognition teams versus teams without recognition.

When teams with intent recognition agents competed against teams without recognition agents (Figure 7.3) the teams with intent recognition agents won more matches in three of the six datasets.

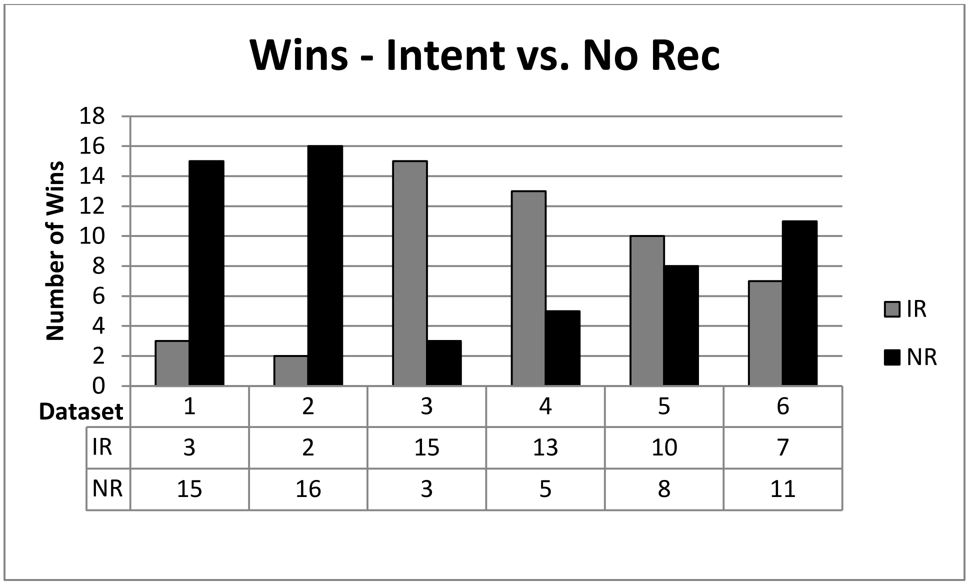
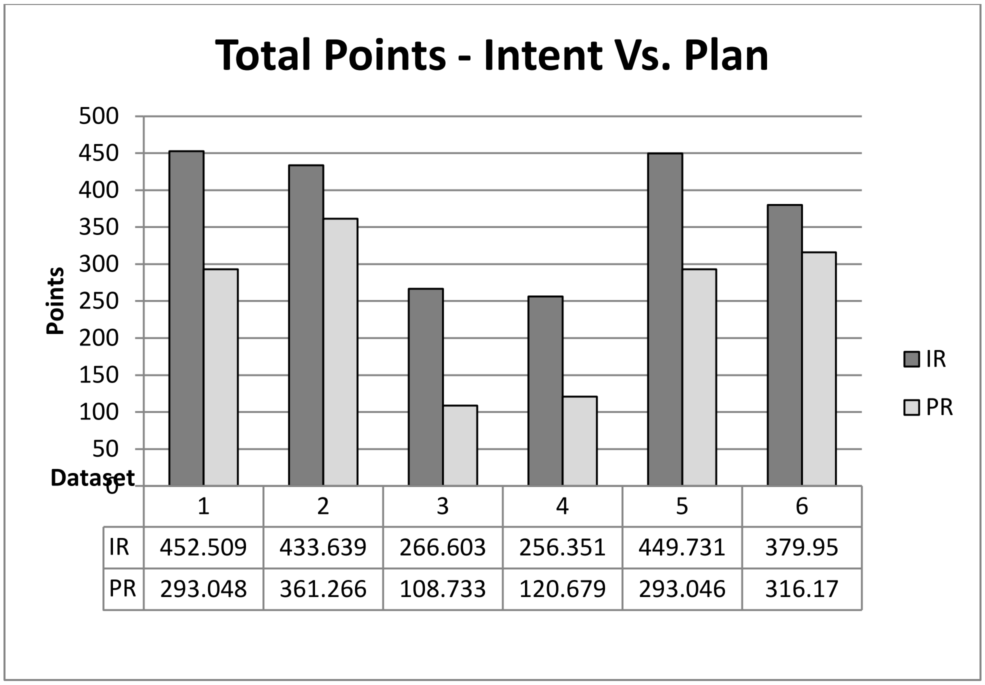


Figure 7.3: Number of wins for intent recognition teams versus teams without recognition.

7.1.2 Overall Score

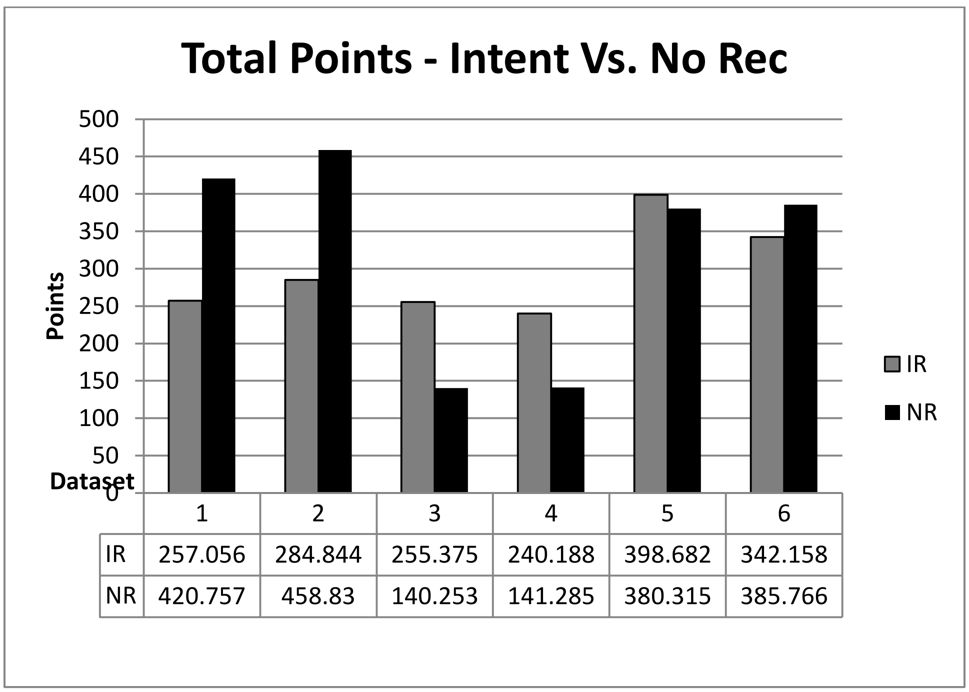
For the following figures (Figure 7.4, Figure 7.5, and Figure 7.6), all of the points, representing the number of cows that were herded per timestep, as scored for each dataset, were added together for the purposes of comparison. Point totals were calculated in the same manner as the MAPC. As shown in Figure 7.4, the teams with intent recognition outscored the teams with plan recognition in every dataset.

As shown in Figure 7.5, the teams with intent recognition outscored the teams without recognition in three of the six datasets. We will explore the differences between datasets and analyze how they relate to the total points in a later section.



	1	2	3	4	5	6
IR	452.509	433.639	266.603	256.351	449.731	379.95
PR	293.048	361.266	108.733	120.679	293.046	316.17

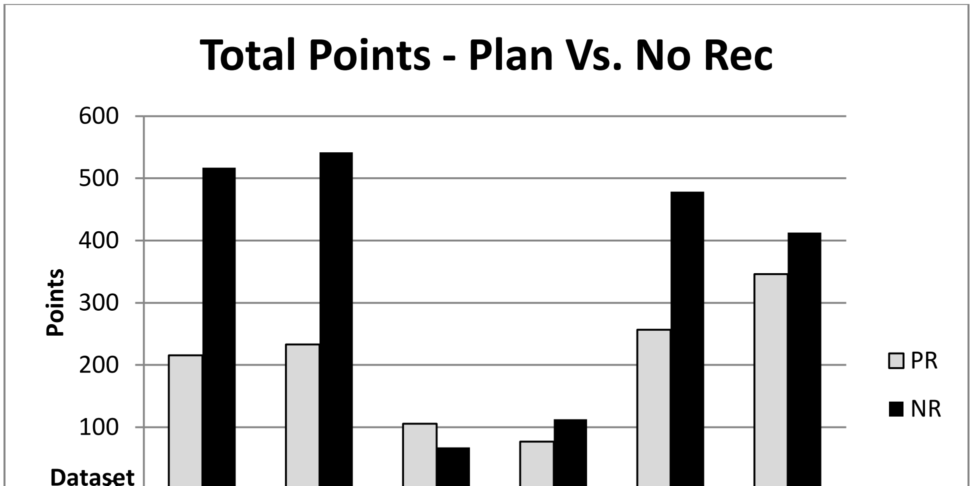
Figure 7.4: Total Points scored for intent recognition teams versus plan recognition teams.



	1	2	3	4	5	6
IR	257.056	284.844	255.375	240.188	398.682	342.158
NR	420.757	458.83	140.253	141.285	380.315	385.766

Figure 7.5: Total points scored for intent recognition teams versus teams without recognition.

As shown in Figure 7.6, in terms of total points scored, only in dataset 3 did the teams with plan recognition have a higher total point value than teams without recognition. Dataset 3 was unique in that it had a short "time to recognition" and the observed agents were following an inefficient plan. This result is further explored in the later sections on "time to communication" and "plan followed."



Dataset	1	2	3	4	5	6
PR	215.769	233.246	105.542	76.872	256.934	346.038
NR	517.347	541.896	67.838	112.759	478.481	412.908

Figure 7.6: Total points scored for plan recognition teams versus teams without recognition.

7.1.3 Differences Between Team Types

A t-test was performed to determine if the points scored by intent recognition agents were statistically different than the points scored by plan recognition, with α set to .05. Across all six datasets, the mean score of teams with intent recognition was found to be statistically higher than the mean score of teams with plan recognition.

The data from all six datasets were combined to determine whether the points scored by plan recognition agents, and the total points scored by intent recognition agents were statistically different than no recognition. Further analysis of factors such as the number of agents per team and the plan that the observed agent is following will follow later.

Teams with no recognition had statistically higher means for points scored than teams with plan recognition when results from all datasets were combined. Additionally, we were unable to find a statistical difference between the scores of the teams with intent recognition and the teams with no recognition when results from all datasets were combined.

7.2 Recognition

Several aspects of recognition were taken into account when interpreting the results. The number of recognition agents on the team was varied across the experiments to determine what the effect would be. The plan that the observed agents followed was varied, which affected what the recognition agents observed and recognized. The plans that were recognized by the recognition agents were recorded and analyzed.

7.2.1 Number of Recognition Agents Per Team

For each of the teams which included recognition agents, the number of recognition agents in the team was varied. Experiments were run with 5, 10, and 15 recognition agents on the team. The rest of the 20 agent team consisted of observed (or non-recognition) agents. The value of α was set to .05 for these tests.

When comparing teams with 5, 10, or 15 intent recognition agents to teams with an equal number of plan recognition agents, the scores of the teams with intent recognition agents had statistically higher means.

When comparing teams with 5, 10, or 15 plan recognition agents to teams with no recognition agents, the scores of the teams without recognition had a statistically higher mean.

When comparing teams with 5, 10, or 15 intent recognition agents to teams with no recognition agents, the mean scores of the two teams were not statistically different.

These comparisons are displayed in Table 7.1. The first column of the table specifies which two team types were competing. The order of the teams is not significant. IR represents teams with intent recognition agents, PR represents teams with plan recognition agents, and NR represents teams without recognition.

Competing Teams	Recognition Agents	Statistically Different	Team With Higher Mean Score
IR, PR	5	Y	IR
IR, NR	5	N	-
PR, NR	5	Y	NR
IR, PR	10	Y	IR
IR, NR	10	N	-
PR, NR	10	Y	NR
IR, PR	15	Y	IR
IR, NR	15	N	-
PR, NR	15	Y	NR

Table 7.1: Comparison of team scores based on the number of recognition agents on each team.

7.2.2 Plan followed by Observed Agents

In datasets 1 through 4, it is known which plan the observed agents are following. In datasets 1 and 2, the observed agents are following Plan 2. In datasets 3 and 4, the observed agents are following Plan 4. In datasets 5 and 6, the observed agents choose a random plan to follow. The observed agents randomly chose their plan and were equally likely to choose an “effective,” “ineffective,” or “average” plan.

Plan 2 is modeled after the JIAC V team by Heβler *et al.* (2010), which won the Multi-Agent competition in 2009. This is used as a known winning strategy. In this plan, agents herd individual cows into their team's corral. Agent's select cows to herd based on both the agent's distance to the goal and the cow's distance to the goal.

Plan 4 is used as an example of a weak strategy. It consists of herding the cows together in big groups but not necessarily towards either corral.

Regardless of the plan that the observed agents were following, whether it was Plan 2 (the "best" plan), Plan 4 (the "poor" plan), or a random plan (to simulate an "average" group performance), when comparing teams with intent recognition agents to teams with plan recognition agents, the mean scores of the two teams were statistically different, with the mean score of the teams with intent recognition agents being higher in all cases. We compared the team scores, and the results are displayed in Table 7.2. The first column specifies which two team types were competing. The order in which the teams are listed is not significant. IR represents teams with intent recognition agents, PR represents teams with plan recognition agents, and NR represents teams without recognition.

Competing Teams	Observed Agents' Plan	Statistically Different	Team With Higher Mean Score
IR, PR	2	Y	IR
IR, NR	2	Y	NR
PR, NR	2	Y	NR
IR, PR	4	Y	IR
IR, NR	4	Y	IR
PR, NR	4	N	-
IR, PR	Random	Y	IR
IR, NR	Random	N	-
PR, NR	Random	Y	NR

Table 7.2: Comparison of team scores based on the plan that the observed agents were following.

In the scenario when observed agents were following Plan 2 (the "best" plan), when comparing teams with plan recognition agents to teams without recognition agents, the mean scores of the two teams were statistically different. The mean score of the teams with plan recognition agents was lower. In the scenario when observed agents were following Plan 4 (the "poor" plan), the mean scores of the two teams were not statistically different. In the scenario when observed agents were following a randomly chosen plan (to simulate an "average" group performance), the mean scores of the two teams were statistically different, with the mean score of the teams with non-recognition agents being higher.

Non-recognition agents outperformed teams with plan recognition only when the observed agents were following the "best" plan and when they were following a random plan. When the observed agents were following a "poor" plan, the results were not statistically different.

In the scenario when observed agents were following Plan 2 (the "best" plan), when comparing teams with intent recognition agents to teams without recognition agents, the mean scores of the two teams were statistically different. The mean score of the teams with intent recognition agents was lower. This is due to the fact that having the entire team following an efficient plan is more effective than having some of the team members follow the efficient plan and the remaining team members perform recognition. However, in the scenario when observed agents were following Plan 4 (the "poor" plan), the mean scores of the two teams were statistically different, with the mean score of the teams with intent recognition agents being higher. In the scenario when observed agents were following a randomly chosen plan (to simulate an "average" group performance), the mean scores of the two teams were not statistically different.

The teams with no recognition outperformed intent recognition when the observed agents were following the "best" plan. Intent recognition outperformed no recognition when the observed agents were following the "poor" plan. The two team types were not statistically different when the observed agents were following the "average" plan.

7.2.3 Recognized Plans

In this section we will examine the accuracy of plan recognition with differing "time to communication" values and different observed plans. Here we will focus on Plan 2 (the near optimum plan) and Plan 4 (the inefficient plan). We also examine how intent recognition agents select from a broad number of plans from the plan library, which confirms our hypothesis H3.

As indicated earlier, in datasets 1 through 4, it is known which plan the observed agents are following. In datasets 1 and 2, the observed agents are following Plan 2. In datasets 3 and 4, the observed agents are following Plan 4. In datasets 5 and 6, the observed agents are following a randomly chosen plan from the plan library of 8 plans.

When observed agents are following Plan 2, as shown in Figure 7.7, with "time to communication" set to 200, and Figure 7.8 with "time to communication" set to 50, plan recognition agents tend to recognize the correct plan as being either plan 2 or plan 4. Intent recognition agents recognize a wider variety of plans. This does not necessarily mean that the intent recognition agents have recognized an incorrect plan. Intent recognition agents attempt to find a plan with the same intent as the observed agent, not necessarily the same exact plan.

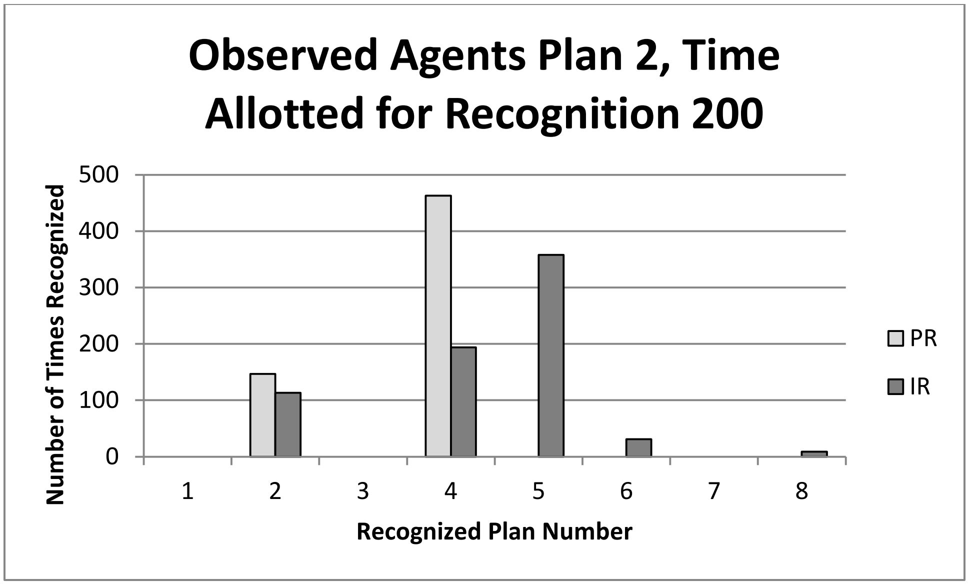


Figure 7.7: Plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following Plan 2 and the allotted time for recognition was 200 timesteps.

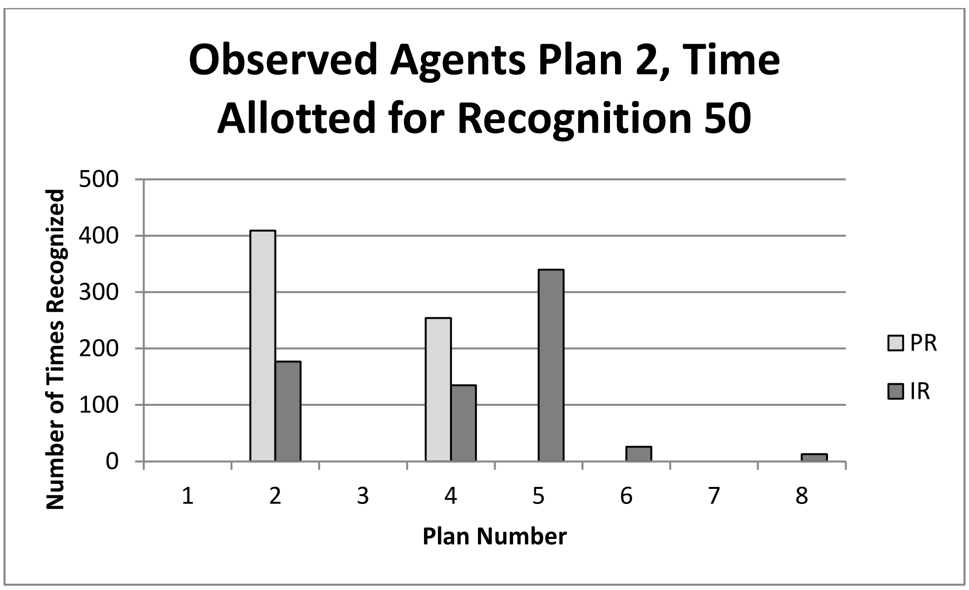


Figure 7.8: Plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following Plan 2 and the allotted time for recognition was 50 timesteps.

When comparing Figure 7.7 and Figure 7.8, it can be seen that plan recognition agents were more accurate when given more time to complete recognition.

As shown in Figure 7.9 and Figure 7.10, when the observed agents are following Plan 4, plan recognition agents correctly recognize this most of the time. However, this is not necessarily helpful to the team because Plan 4 is purposefully chosen as a “poor” plan. So although the plan recognition agents are performing recognition correctly, it does not help in increasing the team score. In contrast, intent recognition agents recognize a wider number of plans to have similar intents. In this way, they are able to select plans that are more effective than the one that the observed agents are following.

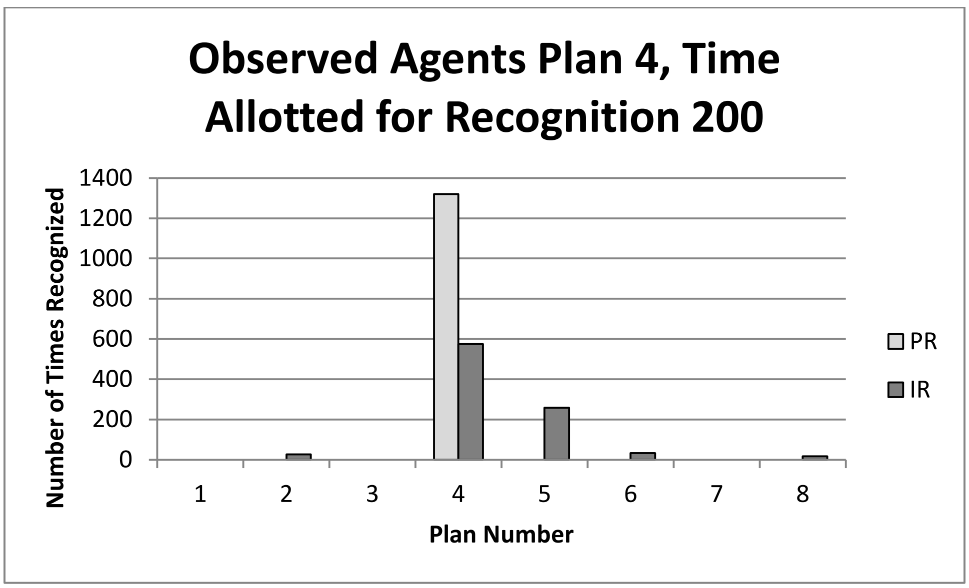


Figure 7.9: Plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following Plan 4 and the allotted time for recognition was 200 timesteps.

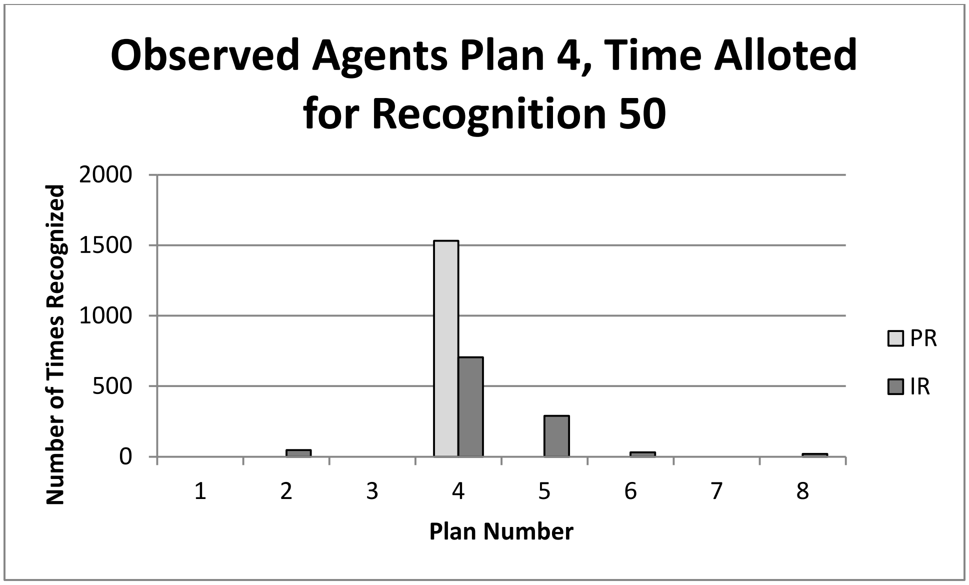


Figure 7.10: Plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following Plan 4 and the allotted time for recognition was 50 timesteps.

Figure 7.11 and Figure 7.12 show the plans selected by plan recognition and intent recognition agents when the observed agents were following a randomly selected plan.

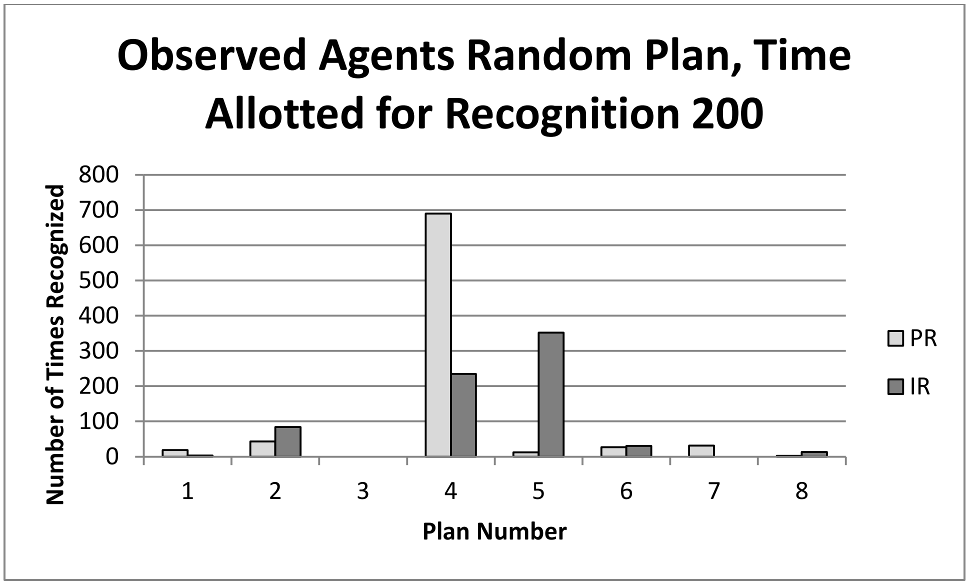


Figure 7.11: Experiment 2. The plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following a random plan from the plan library and the allotted time for recognition was 200 timesteps.

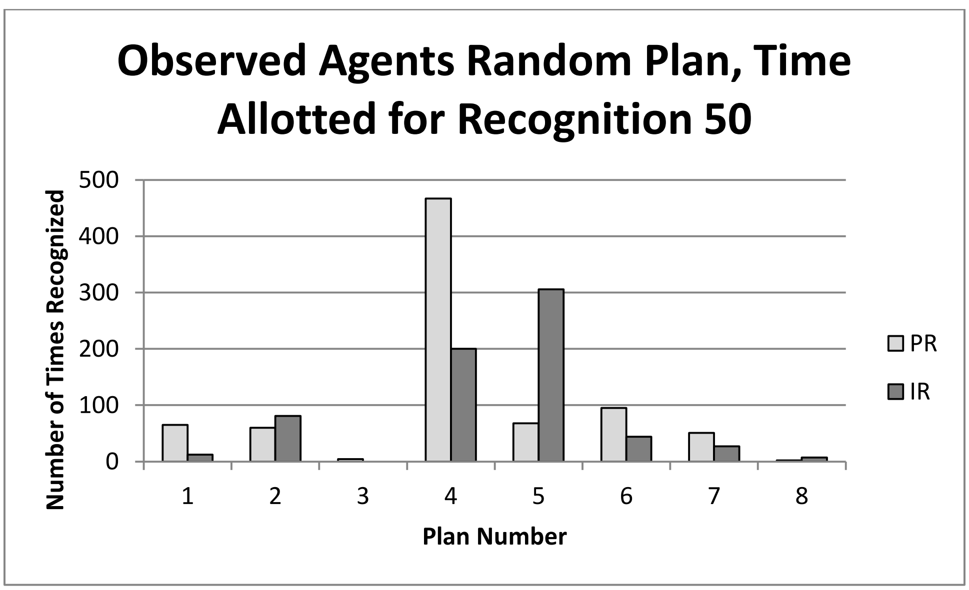
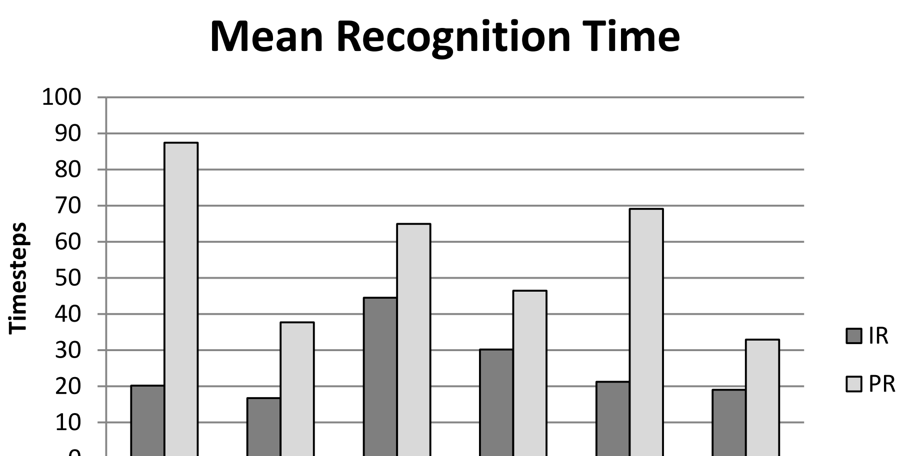


Figure 7.12: Experiment 2. The plans recognized by plan recognition agents and intent recognition agents in the case where the observed agents were following a random plan from the plan library and the allotted time for recognition was 50 timesteps.

7.2.4 Mean Recognition Time

Recognition time is the number of timesteps an agent takes to complete the process of recognition. The mean recognition time was recorded for all intent versus plan recognition competitions, an overview of which can be seen in Figure 7.13. The mean recognition time for intent recognition was statistically lower than the mean recognition for every dataset. Intent recognition is faster, and therefore more efficient, at plan selection in all six datasets.



Dataset	1	2	3	4	5	6
IR	20.22282	16.78312	44.51174	30.19979	21.26653	19.05077
PR	87.40903	37.72093	64.96523	46.41812	69.13443	32.9138

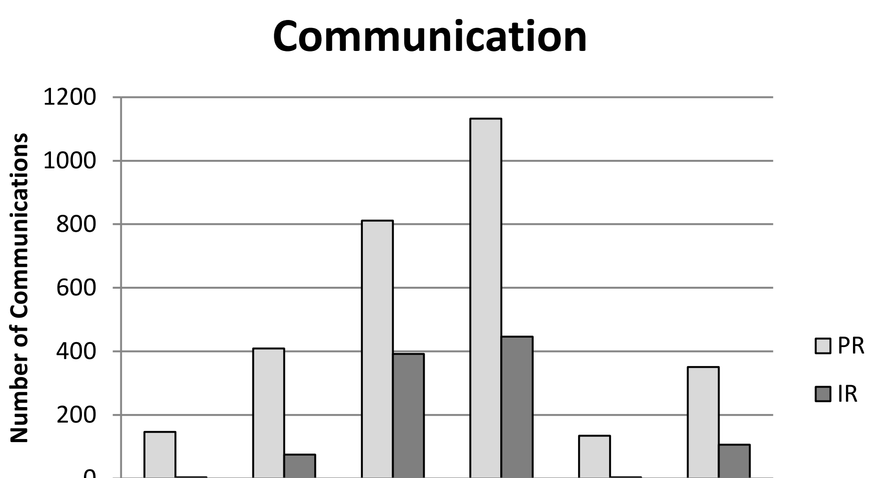
Figure 7.13: The mean recognition time of plan recognition agents and intent recognition agents by dataset. Intent recognition has a statistically lower mean recognition time in all datasets.

7.3 Communication

As mentioned previously, one of the primary purposes of recognition is to reduce the amount of communication in a multi-agent system. Both plan recognition agents and intent recognition agents have the ability to communicate. In this section, we investigate the number of communications and the time to communication.

7.3.1 Number of Communications

In every dataset, the plan recognition agents communicated a greater number of times than the intent recognition agents (Figure 7.14). This confirms our hypothesis H4.



Dataset	1	2	3	4	5	6
PR	147	409	811	1132	135	351
IR	4	75	392	446	4	106

Figure 7.14: The number of times plan recognition agents and intent recognition agents communicated per dataset.

7.3.2 Time to Communication

In this section we examine how allotting more or less time for recognition effects plan recognition and intent recognition agents.

In datasets 1, 3, and 5, recognition agents had a time limit of 200 timesteps to complete recognition before they communicated. In datasets 2, 4, and 6, recognition agents had a time limit of 50 timesteps before they would communicate.

In the scenario where recognition agents were given time allotment of 50 timesteps to complete recognition, when comparing teams with intent recognition agents to teams with plan recognition agents, the mean scores of the two teams were statistically different. The mean score of the teams with intent recognition agents was higher. A similar result was found when these two teams competed with a time allotment of 200, as shown in Table 7.3. The first column specifies which two team types were competing. The order is not significant. IR represents teams with intent recognition agents, PR represents teams with plan recognition agents, and NR represents teams without recognition.

Competing Teams	Time To Recognition	Statistically Different	Team With Higher Mean Score
IR, PR	50	Y	IR
IR, NR	50	N	-
PR, NR	50	Y	NR
IR, PR	200	Y	IR
IR, NR	200	N	-
PR, NR	200	Y	NR

Table 7.3: Comparison of team scores based on the time allotment for recognition.

In the scenario where recognition agents were given time allotment of 50 timesteps to complete recognition, when comparing teams with plan recognition agents to teams without recognition agents, the mean scores of the two teams were statistically different. The mean score of the teams without recognition agents was higher. This was the same result when these two teams competed with a time allotment of 200.

In the scenario where recognition agents were given time allotment of 50 timesteps to complete recognition, when comparing teams with intent recognition agents to teams without recognition agents, the mean scores of the two teams were not statistically different. This was the same result when these two teams competed with a time allotment of 200.

Therefore we conclude that "time to communication" is not a significant factor in which team wins. This may be because most of the recognition is completed before 50 timesteps so increasing the limit has little effect on the results. Regardless of the "time to communication," when intent recognition and plan recognition teams competed, intent recognition won more often. When intent recognition and no-recognition teams competed, there was no statistical difference. When plan recognition and no-recognition teams competed, no-recognition teams won more often. This relates to the previous section detailing mean recognition time.

7.4 Adding "Helper" Agents

Although having recognition agents as part of a team is one possible scenario, one of the rationales behind intent recognition is that they are agents that are able to seek out and aid other agents. For this reason, additional trials were run with two teams of 20 agents with no recognition agents. In each run, one of the teams would have one or more additional "helper" agents. In this way, we were able to test whether adding a plan or intent recognition agent to a team would help or hinder the team. We were also able to test whether it would be more beneficial to add a plan, intent, or another non-recognition agent to the team.

Adding one additional agent does not make a statistical difference to the score in any of the cases, regardless of agent type. The charts with statistical details are included in the Appendix.

Adding five additional agents makes a statistical difference in most of the cases. When five intent recognition or non-recognition agents were added, the mean score was statistically higher than the team without this assistance. It was only the situation where plan recognition agents were added where we did not see an advantage. Tables detailing the statistical details are included in the Appendix, along with charts detailing the statistical analysis.

Chapter 8 Conclusion

This research defined and described intent recognition in multi-agent systems. We designed and built a simulation environment to analyze this concept. We compared intent recognition systems to plan recognition systems in order to study the merits of intent recognition.

Our key research questions were: (1) What are intent recognition systems? (2) How can they be used to have agents autonomously assist each other?

We found that intent recognition agents are able to utilize unobserved actions to increase the overall utility of the system. We also found that intent recognition agents minimize communication, and therefore preserve resources for when they are most needed. This reduction of communication is also useful in adversarial systems when communications may be intercepted. Intent recognition is an important construct in multi-agent systems because, as this research has shown, it can be used to increase the utility of a variety of multi-agent systems while minimizing the amount of communication.

The intent recognition framework is applicable in many domains. An example of this would be in robotics, particularly if the robots are located in space. The robots could be cooperatively working on a task in space, such as space station assembly. There may be cameras in certain locations to monitor the work, but people on Earth may not have a way to view every robot simultaneously. Because of this, people may not be able to determine if robots are working inefficiently at their tasks, are in need of assistance, or are in need of repair. This is particularly true if a robot's communication systems are damaged and it is not able to communicate with people on Earth or with the surrounding robots. If the robots in this cooperative system are able to perform intent recognition, they would be able to determine what the damaged robot is attempting to achieve. These intent recognition robots would then be able to aid the damaged robot in its task or even repair the robot, depending on their capabilities.

Incorporating intent recognition in future multi-agent systems will lead to agents that are more efficient, have fewer dependencies on human interaction, and are one step closer to accurately emulating their human counterparts.

In our research, a series of hypotheses were tested.

8.1 Analysis of Hypotheses: Box Pushing

The following is a summary of our hypothesis testing for the Box Pushing Experiment, an outline of which can be seen in Table 8.1.

H1.1 Intent recognition systems will be able to reduce task completion time when compared with plan recognition systems.

We partially confirmed H1.1 in our research. There were cases where intent recognition systems had a significantly lower completion time than plan recognition systems.

It is our presumption that in cases where the observed agents were following a more complex plan, the plan recognition agents spent a greater amount of time than the intent recognition agents on the recognition phase. On the other hand, when observed agents were following a simple plan, plan and intent recognition agents spent a similar amount of time on the recognition phase.

H2.1 Intent recognition systems will be able to increase percent of task completion when compared with plan recognition systems.

H2.1 was rejected in our research. While intent recognition agents may be faster at completing tasks than plan recognition agents, the increase in potential for error makes the two approaches comparable in terms of completion percentage.

H3 Intent recognition systems will use unobserved actions to complete the given task.

As hypothesized, in our research plan recognition agents and intent recognition agents approached the task of aiding fellow agents in different ways, confirming H3. Unlike agents performing plan recognition, intent recognition agents have the ability to dynamically choose actions to assist other agents in the system using only the knowledge in their plan libraries.

H4 Intent recognitions will communicate fewer times compared with plan recognition systems.

We confirmed H4 in our research. While both approaches aim to reduce communication in a system, agents performing intent recognition have the advantage of not trying to make an exact match between observations and the plan library. Thus, communication is further reduced in the case where all agents in the system have differing capabilities and plan libraries.

Hypothesis	**Results**
H1.1	Partially Confirmed
H2.1	Rejected
H3	Confirmed
H4	Confirmed

Table 8.1: Box Pushing Hypothesis Testing Results.

8.2 Analysis of Hypotheses: Cow Herding

The following is a summary of our hypothesis testing for the Cow Herding Experiment, an outline of which can be seen in Table 8.2.

Hypothesis	Results
H1.2	Confirmed
H2.2	Confirmed
H3	Confirmed
H4	Confirmed

Table 8.2: Cow Herding Hypothesis Testing Results.

H1.2: Teams which include agents using intent recognition systems will be able to score more overall points than teams which include agents using plan recognition systems.

We confirmed H1.2 in our research. Teams with intent recognition scored a higher total number of points in every data set when compared to plan recognition systems. Over all six datasets, intent recognition teams scored a total of 2,238.783 points and plan recognition systems scored a total of 1,492.942 points. Details of how points were calculated were discussed in a previous section.

H2.2: Teams which include agents using intent recognition systems will win a greater number of games than teams which include agents using plan recognition systems.

H2.2 was confirmed in our research. Teams with intent recognition agents won more games than teams with plan recognition agents in every dataset. When these two team types were directly competing, teams with intent recognition agents won a total of 88 games, while teams with plan recognition agents won 20 games.

H3. Intent recognition systems will use unobserved actions to complete the given task.

We confirmed H3 in our research. In datasets 1 - 4 the plan that the observed agents followed was known. Intent recognition agents recognized plans aside from the known plan.

H4. Intent recognitions will communicate fewer times compared with plan recognition systems.

H4 was confirmed in our research. There were fewer communications by intent recognition agents than plan recognition agents in every dataset. Across all six datasets, intent recognition agents communicated 1,027 times and plan recognition agents communicated 2,985 times.

8.3 Overall Results

Our hypothesis testing indicates that in less complex scenarios, such as box pushing, adding recognition agents to the system, whether they are performing plan recognition or intent recognition, increases the overall utility of the system. The measure of utility varies by domain, but in box pushing it can be measured by the percent of the task that was completed and the time in which tasks were completed. In more complex scenarios, such as cow herding, intent recognition has an advantage in some situations. If the observed agents are highly organized and efficient, then there is not a great gain from taking the time and resources to perform recognition. However, if the observed agents are not organized or are performing tasks inefficiently, there is an advantage to receiving assistance from agents that are performing intent recognition over agents that are performing plan recognition.

Based on the fact that there is an advantage in using intent recognition, it is imperative that people who are designing cooperative multi-agent systems consider the intent recognition concept. With a properly seeded plan library, idle agents are able to increase the utility of a system by aiding other agents without human intervention.

8.4 Theoretical Contribution

This work makes multiple theoretical contributions. Its theoretical contributions are primarily in the field of multi-agent systems.

1. We designed, developed, and tested a novel concept called intent recognition, where agents determine the intent of the agents around them.
2. Agents performing intent recognition have a reduction in the number of communications. This reduces the overall communication in a system.
3. With a properly generated plan library, adding agents which perform intent recognition to a system can increase the overall utility, where the utility is domain specific.
4. Intent recognition allows agents to autonomously find ways in which to utilize their idle time. This reduces the amount of time agents are waiting for instructions. It also increases utility of the system by having all agents working at all times.
5. Intent recognition allows agents to assist each other using unobserved actions. Unlike agents performing plan recognition, agents performing intent recognition search for plans in their libraries which have the same goal as the observed actions instead of the same steps. This leads to agents that are able to use actions to achieve the goal that the observed agent may not be capable of.

8.5 Limitations

Our experiments use a single straightforward plan recognition system for comparison to the proposed intent recognition system. Comparison to other plan recognition systems was not performed.

As shown in our research, the utility of agents using intent recognition can be domain and capability specific. Our experiments were run in two specific domains, box pushing and cow herding.

Another constraint in our system was that the variable referred to as “intent bonus” was kept constant in the interest of keeping a manageable number of variables.

Intent recognition agents select plans which they deem to have a similar intent to the plan that they are observing, while not necessarily the exact same plan steps. Using these unobserved actions, agents using intent recognition can be beneficial to the overall system goal. However, these outside actions can also introduce errors which can be an impediment to the intent of the observed agents. This can occur when the intent of the recognized plan is slightly different than the actual observed intent.

8.6 Future Work

In our future research, we plan to incorporate intent recognition into more complex variations of the box pushing scenario. For example, we will give different weights to the boxes. Some plans in the intent recognition agent's plan library will adjust for weight and others will not. We will also add agents with different capabilities. A fueling agent can refill the energy of an agent that has stopped. We will vary action costs in order to study their effects on the effectiveness of intent recognition. Plans will be added to the plan library and the observed agents will be given more complex starting plans to follow.

We will also expand our experiments in the cow herding domain. We will vary the number of cows on the field. We will adjust more of the variables, such as the weights that are assigned to grid locations containing various objects. We will also test more values of the "time to communication" variable, including a very high value which represents not having a time limit. Complex actions will be added which will then be used to build a greater variety of plans in the plan library. We will run experiments with new starting plans for the observed agents.

We also plan to compare intent recognition systems to different plan recognition systems. We will also test various ways for intent recognition agents to compare the similarity of their observations to the plans in their plan library.

We will vary and study the effects of the intent bonus to examine its effect on intent recognition. We will also study ways to reduce the amount of error introduced into the system by incorrect assumptions made by the intent recognition agent.

Intent recognition will be tested in new and more complex domains, such as the Agents on Mars scenario proposed by the MAPC.

Appendix

Experiment Series 1:

Dataset	*R Square*	*Adjusted R Square*	*ANOVA F*	*Significance F*	*Coefficient*	*t Stat*	*P-value*
1	0.704275	0.697477219	103.5963602	9.62939E-24	-19.05	-3.80072	0.000267
2	0.827705	0.823651427	204.1705532	3.47972E-33	-17.096652	-3.71432	0.000364
3	0.086875	0.064603471	3.900743885	0.024084805	13.4348441	2.393831	0.018956
4	0.178191	0.15862366	9.106737815	0.000263262	8.58877067	1.622191	0.10851
5	0.119711	0.099239601	5.84761811	0.004157936	16.5666667	2.978746	0.003761
6	0.382356	0.367650441	26.00036375	1.62554E-09	33.3113917	5.78711	1.2E-07
7	0.369957	0.355132484	24.95571599	2.97326E-09	32.862069	5.517419	3.65E-07
8	0.194476	0.175958234	10.50211717	8.21133E-05	23	4.438913	2.64E-05
9	0.269988	0.252811725	15.71825835	1.55422E-06	-13.414118	-1.26054	0.210925
10	0.162256	0.142997989	8.425199035	0.000452202	36.35	3.954091	0.000156
11	0.743468	0.737570299	126.0692212	1.98466E-26	-149.51667	-7.22494	1.81E-10

Table 0.1: Experiment Series 1. Regression Results for time and recognition type. Compares no recognition, plan recognition, and intent recognition.

Dataset	*R Square*	*Adjusted R Square*	*ANOVA F*	*Significance F*	*Coefficient*	*t Stat*	*P-value*
1	0.71411514	0.70408409	71.19048	3.17284E-16	-38.766667	-4.24379	8.18E-05
2	0.81337457	0.806709377	122.0331	3.8653E-21	-26.974936	-2.876	0.005687
3	0.55379018	0.536952079	32.8891	5.16163E-10	-54.64074	-6.90339	6.45E-09
4	0.6324311	0.619064953	47.3159	1.11382E-12	-48.101547	-6.23671	6.65E-08
5	0.53234949	0.515647686	31.87377	5.72532E-10	-46.452344	-6.00249	1.51E-07
6	0.47653695	0.457501933	25.03475	1.85937E-08	-37	-4.50266	3.53E-05
7	0.45343291	0.433912661	23.22884	4.50835E-08	-37.67729	-4.2553	8.02E-05
8	0.14782039	0.117919348	4.943654	0.010474601	-21.866667	-2.20597	0.031432
9	0.60536539	0.591518563	43.7187	3.10143E-12	-103.4	-5.67638	4.84E-07
10	0.05915922	0.026147261	1.792054	0.175877314	-35.533333	-1.80754	0.075956
11	0.79108405	0.78375367	107.9185	4.16093E-20	-40.866667	-1.49934	0.139303

Table 0.2: Experiment Series 1. Regression results for dependent variable time. Independent variable is recognition type, comparing intent recognition and plan recognition.

Dataset	*R Square*	*Adjusted R Square*	*ANOVA F*	*Significance F*	*Coefficient*	*t Stat*	*P-value*
1	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3	0.752823	0.747007	129.4418	1.6E-26	0.084667	5.279956	9.77E-07
4	0.695799	0.688725	98.35392	5.97E-23	0.078255	4.669698	1.1E-05
5	0.70048	0.693595	101.7326	1.68E-23	0.115333	6.451828	6.02E-09
6	0.835878	0.832062	219.0009	1.79E-34	0.062988	5.72825	1.46E-07
7	0.829383	0.825461	211.4571	3.92E-34	0.028833	5.396981	5.78E-07
8	0.776796	0.771665	151.389	4.66E-29	0.051111	6.605111	3.03E-09
9	0.658198	0.650341	83.76682	5.24E-21	0.030278	1.91934	0.058219
10	0.830521	0.826625	213.1694	2.93E-34	0.082963	7.473222	5.75E-11
11	0.05413	0.032386	2.489417	0.088851	-0.00062	-1.678	0.096936

Table 0.3: Experiment Series 1. Regression for dependent variable percent. All recognition types represented.

Dataset	*R Square*	*Adjusted R Square*	*ANOVA F*	*Significance F*	*Coefficient*	*t Stat*	*P-value*
1	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3	0.671111	0.659151	56.11472	5.23E-14	0.00933	0.273492	0.785499
4	0.59994	0.585653	41.98957	7.24E-12	-0.01148	-0.34854	0.728742
5	0.590351	0.575977	41.07175	8.99E-12	0.004	0.110685	0.912255
6	0.878374	0.874031	202.2145	2.4E-26	0.005531	0.276715	0.783018
7	0.892807	0.889046	237.3748	2.29E-28	-0.00833	-0.91438	0.36437
8	0.713945	0.703908	71.13121	3.23E-16	0.037222	2.261641	0.027552
9	0.587531	0.573059	40.59618	1.09E-11	-0.01778	-0.58155	0.563162
10	0.783415	0.775816	103.0883	1.16E-19	0.069259	2.716663	0.008716
11	0.066568	0.033817	2.032502	0.140395	-0.00125	-1.37427	0.174739

Table 0.4: Experiment Series 1. Regression results for dependent variable percent. Intent recognition and plan recognition are represented.

Dataset			1			2			3			4			5			6	
Recognition Type		NR	PR	IR	NR	PR	IR	NR	PR	IR	NR	PR	IR	NR	PR	IR	NR	PR	IR
	5	364.8	330	286.2	411	402.6	376	198.4	309.6	236.4	208.6	315.6	234	182.8	303.8	236.2	195.2	296.8	208.8
	10	228.8	252.8	192.8	340	316.2	279.6	204.2	315	256.8	205.8	312	296	201.2	321	303	204	302.2	247.6
Population Size	15	215.2	209.2	154.2	264	248	236	211.4	329.4	297.4	212.4	309	257.4	214.4	307	275.4	195.8	306	278.8
	20	188.4	179.4	140.6	218.4	230.8	187.2	228.4	289.6	247.4	217.6	268.6	222.2	211	267	240.6	215	323	284.2
	25	154.6	171.2	144.6	201.4	197.8	181.6	226.8	273.8	212.4	211.4	246.6	194.6	230.8	271.4	221.2	214.2	325.8	294.8
	30	133.6	146.8	138.4	175.2	183.2	150.8	225	265.6	210.8	232.4	233	190	231.8	272.6	194.4	227	327.2	325.2
Dataset			7			8			9			10			11				
Recognition Type		NR	PR	IR	NR	PR	IR	NR	PR	IR	NR	PR	IR	NR	PR	IR			
	5	172.6	271.6	233.2	203.6	319.2	253.2	413.6	542.4	418.2	417.6	488.2	448.6	1685	1142	1046			
	10	196	309.6	223.8	211.6	308.8	260.6	434.8	656.4	492	442.2	658	517.4	1035	770	718.2			
Population Size	15	211.6	317.4	279	224.8	300	268.8	450.8	589.4	429.4	447.8	575.8	511.6	869.6	633	637.2			
	20	223.2	320	296.6	218.8	306.8	264	441	508.8	383.6	459.8	550.2	595.8	753.2	569.2	522.2			
	25	224.8	326.6	303.4	225.8	246.4	264.6	412.8	377.4	370.8	456.2	549.4	545.4	659.6	485.4	507.2			
	30	221.6	322.2	311.8	228	238.6	277.4	410.6	355.6	315.6	463.2	514.6	504.2	628.6	482.4	406.4			

Table 0.5: Experiment Series 1. Average times across all datasets by recognition type and population size.

Set	Actual	Recognized Plan	Count	Help	Error
1	YBE	PushYellowBlueBoxesEastPlan	52	Y	Y
		PushYellowBoxesEastPlanDF	275	Y	N
		PushYellowGreenBoxesEastPlan	28	Y	Y
2	YBE	PushYellowBlueBoxesEastPlan	47	Y	Y
		PushYellowBoxesEastPlan	12	Y	N
		PushYellowBoxesEastPlanDF	326	Y	N
		PushYellowGreenBoxesEastPlan	40	Y	Y
3	YBE	PushYellowBlueBoxesEastPlan	23	Y	Y
		PushYellowBoxesEastPlan	34	Y	N
		PushYellowBoxesEastPlanDF	429	Y	N
		PushYellowGreenBoxesEastPlan	25	Y	Y
4	YBE	PushYellowBlueBoxesEastPlan	39	Y	Y
		PushYellowBoxesEastPlan	53	Y	N
		PushYellowBoxesEastPlanDF	378	Y	N
		PushYellowGreenBoxesEastPlan	24	Y	Y
5	YBE	PushYellowBlueBoxesEastPlan	43	Y	Y
		PushYellowBoxesEastPlan	108	Y	N
		PushYellowBoxesEastPlanDF	316	Y	N
		PushYellowGreenBoxesEastPlan	36	Y	Y
6	YBE	PushYellowBlueBoxesEastPlan	11	Y	Y
		PushYellowBoxesEastPlan	75	Y	N
		PushYellowBoxesEastPlanDF	422	Y	N
		PushYellowGreenBoxesEastPlan	17	Y	Y
7	YEBN	PushYellowBlueBoxesEastPlan	17	Y	Y
		PushYellowBoxesEastPlan	77	Y	N
		PushYellowBoxesEastPlanDF	410	Y	N

		PushYellowGreenBoxesEastPlan	21	Y	Y
8	YEBN	PushBlueBoxesNorthPlan	40	Y	N
		PushBlueBoxesNorthPlanDF	34	Y	N
		PushYellowBlueBoxesEastPlan	31	Y	Y
		PushYellowBlueBoxesNorthPlan	7	Y	Y
		PushYellowBoxesEastPlan	66	Y	N
		PushYellowBoxesEastPlanDF	330	Y	N
		PushYellowGreenBoxesEastPlan	17	Y	Y
9	YEBN	PushAllBoxesEastPlan	3	Y	Y
		PushBlueBoxesNorthPlan	72	Y	N
		PushBlueBoxesNorthPlanDF	318	Y	N
		PushYellowBlueBoxesEastPlan	38	Y	Y
		PushYellowBlueBoxesNorthPlan	15	Y	Y
		PushYellowBoxesEastPlan	5	Y	N
		PushYellowBoxesEastPlanDF	48	Y	N
		PushYellowGreenBoxesEastPlan	26	Y	Y
10	YEBNRW	PushAllBoxesNorthPlan	5	Y	Y
		PushBlueBoxesNorthPlan	18	Y	N
		PushBlueBoxesNorthPlanDF	155	Y	N
		PushRedBoxesEastPlanDF	4	N	Y
		PushRedBoxesWestPlan	46	Y	N
		PushRedBoxesWestPlanDF	155	Y	N
		PushRedGreenBoxesWestPlan	59	Y	Y
		PushYellowBlueBoxesEastPlan	25	Y	Y
		PushYellowBlueBoxesNorthPlan	5	Y	Y
		PushYellowBoxesEastPlanDF	12	Y	N
		PushYellowGreenBoxesEastPlan	41	Y	Y
11	ALL E	PushAllBoxesEastPlan	143	Y	N
		PushRedGreenBoxesEastPlan	195	Y	N
		PushRedGreenBoxesSouthPlan	2	N	Y
		PushRedGreenBoxesWestPlan	9	N	Y
		PushYellowBlueBoxesEastPlan	63	Y	N
		PushYellowGreenBoxesEastPlan	93	Y	N
		Total:	5418	5403	710

Table 0.6: Experiment Series 1: Recognition choices by intent recognition agents.

Experiment Series 2:

t-Test: Two-Sample Assuming Equal Variances		
	Intent	*Plan*
Mean	20.72946699	13.82352
Variance	52.69887933	47.5208
Observations	108	108
Pooled Variance	50.10983747	
Hypothesized Mean Difference	0	
Df	214	
t Stat	7.169001096	
P(T<=t) one-tail	6.04648E-12	
t Critical one-tail	1.652005156	
P(T<=t) two-tail	1.2093E-11	
t Critical two-tail	1.971111258	

Table 0.7: Experiment Series 2. t-Test for difference in means of intent recognition vs. plan recognition.

t-Test: Two-Sample Assuming Equal Variances No Recognition vs Plan Recognition		
	Norec	*Plan*
Mean	19.73360686	11.42964
Variance	146.1770838	43.68479
Observations	108	108
Pooled Variance	94.93093517	
Hypothesized Mean Difference	0	
Df	214	
t Stat	6.2629446	
P(T<=t) one-tail	1.02089E-09	
t Critical one-tail	1.652005156	
P(T<=t) two-tail	2.04179E-09	
t Critical two-tail	1.971111258	

Table 0.8: Experiment Series 2. t-Test for difference in means of plan recognition vs. no recognition.

t-Test: Two-Sample Assuming Equal Variances Intent Recognition vs No Recognition		
	Intent	*NoRec*
Mean	16.63243883	17.8445
Variance	40.9489421	73.70229
Observations	108	108
Pooled Variance	57.32561554	
Hypothesized Mean Difference	0	
Df	214	
t Stat	-1.176374744	
P(T<=t) one-tail	0.120376034	
t Critical one-tail	1.652005156	
P(T<=t) two-tail	0.240752069	
t Critical two-tail	1.971111258	

Table 0.9: Experiment Series 2. t-Test for difference in means of intent recognition vs. no recognition.

t-Test: Two-Sample Assuming Equal Variances 5 Recognition Agents Per Team - Intent Vs Plan		
	Intent	*Plan*
Mean	19.09802617	14.02074
Variance	62.86226083	49.04453
Observations	36	36
Pooled Variance	55.95339427	
Hypothesized Mean Difference	0	
Df	70	
t Stat	2.879751361	
P(T<=t) one-tail	0.002638291	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.005276582	
t Critical two-tail	1.994437112	

Table 0.10: Experiment Series 2. t-Test for intent recognition vs. plan recognition. 5 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 5 recognition agents per team - Plan Vs No Rec		
	NoRec	*Plan*
Mean	18.2276558	12.01353
Variance	101.3541651	48.15763
Observations	36	36
Pooled Variance	74.75589876	
Hypothesized Mean Difference	0	
Df	70	
t Stat	3.04925444	
P(T<=t) one-tail	0.001618603	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.003237205	
t Critical two-tail	1.994437112	

Table 0.11: Experiment Series 2. t-Test for plan recognition vs. no recognition. 5 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 5 Recognition agents per team - Intent Vs No Rec		
	Intent	*NoRec*
Mean	15.917831	17.10934
Variance	57.27899673	64.30838
Observations	36	36
Pooled Variance	60.7936882	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-0.648340566	
P(T<=t) one-tail	0.259442943	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.518885887	
t Critical two-tail	1.994437112	

Table 0.12: Experiment Series 2. t-Test for intent recognition vs. no recognition. 5 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 10 recognition agents per team - Intent Vs Plan		
	Intent	*Plan*
Mean	20.19466622	14.1792
Variance	49.44405992	50.12656
Observations	36	36
Pooled Variance	49.78530929	
Hypothesized Mean Difference	0	
Df	70	
t Stat	3.617055273	
P(T<=t) one-tail	0.000279156	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.000558312	
t Critical two-tail	1.994437112	

Table 0.13: Experiment Series 2. t-Test for intent recognition vs. plan recognition. 10 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 10 recognition Agents Per Team No Rec Vs Plan		
	NoRec	*Plan*
Mean	19.79357951	12.21196
Variance	161.8411406	42.22057
Observations	36	36
Pooled Variance	102.0308532	
Hypothesized Mean Difference	0	
Df	70	
t Stat	3.184433577	
P(T<=t) one-tail	0.001082863	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.002165726	
t Critical two-tail	1.994437112	

Table 0.14: Experiment Series 2. t-Test for plan recognition vs. no recognition. 10 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 10 recognition agents per team Intent Vs No Rec		
	Intent	*NoRec*
Mean	16.34392326	17.29945664
Variance	41.44165227	83.51855297
Observations	36	36
Pooled Variance	62.48010262	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-0.512874666	
P(T<=t) one-tail	0.304826879	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.609653758	
t Critical two-tail	1.994437112	

Table 0.15: Experiment Series 2. t-Test for intent recognition vs. no recognition. 10 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 15 Recognition Agents - Intent Vs Plan		
	Intent	*Plan*
Mean	22.89570858	13.27063
Variance	40.94317629	45.62222
Observations	36	36
Pooled Variance	43.28269902	
Hypothesized Mean Difference	0	
Df	70	
t Stat	6.207026164	
P(T<=t) one-tail	1.68237E-08	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	3.36474E-08	
t Critical two-tail	1.994437112	

Table 0.16: Experiment Series 2. t-Test for intent recognition vs. plan recognition. 15 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 15 recognition agents - Plan Vs No Rec		
	NoRec	*Plan*
Mean	21.17958527	10.06343
Variance	179.2019448	40.27239
Observations	36	36
Pooled Variance	109.7371657	
Hypothesized Mean Difference	0	
Df	70	
t Stat	4.502088609	
P(T<=t) one-tail	1.30902E-05	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	2.61805E-05	
t Critical two-tail	1.994437112	

Table 0.17: Experiment Series 2. t-Test for plan recognition vs. no recognition. 15 recognition agents.

t-Test: Two-Sample Assuming Equal Variances 15 recognition agents - Intent Vs No Rec		
	Intent	*NoRec*
Mean	17.63556221	19.1247
Variance	24.82023586	74.9443
Observations	36	36
Pooled Variance	49.88226887	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-0.894533472	
P(T<=t) one-tail	0.187051017	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.374102033	
t Critical two-tail	1.994437112	

Table 0.18: Experiment Series 2. t-Test for intent recognition vs. no recognition. 15 recognition agents.

t-Test: Two-Sample Assuming Equal Variances OA following Plan 2		
	Intent	*Plan*
Mean	24.61521402	18.17537
Variance	26.35813611	23.01749
Observations	36	36
Pooled Variance	24.68781425	
Hypothesized Mean Difference	0	
Df	70	
t Stat	5.498828569	
P(T<=t) one-tail	2.92734E-07	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	5.85468E-07	
t Critical two-tail	1.994437112	

Table 0.19: Experiment Series 2. t-Test for intent recognition vs. plan recognition when observed agents were following plan 2.

t-Test: Two-Sample Assuming Equal Variances OA following Plan 2		
	Norec	*Plan*
Mean	29.42343092	12.47261
Variance	46.85834949	17.89611
Observations	36	36
Pooled Variance	32.37723051	
Hypothesized Mean Difference	0	
Df	70	
t Stat	12.63883753	
P(T<=t) one-tail	4.87811E-20	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	9.75622E-20	
t Critical two-tail	1.994437112	

Table 0.20: Experiment Series 2. t-Test for plan recognition vs. no recognition when observed agents were following plan 2.

t-Test: Two-Sample Assuming Equal Variances OA following Plan 2		
	Intent	*NoRec*
Mean	15.55278332	24.43297
Variance	22.00779187	19.52156
Observations	36	36
Pooled Variance	20.76467394	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-8.267908523	
P(T<=t) one-tail	2.93216E-12	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	5.86432E-12	
t Critical two-tail	1.994437112	

Table 0.21: Experiment Series 2. t-Test for intent recognition vs. no recognition when observed agents were following plan 2.

t-Test: Two-Sample Assuming Equal Variances OA following plan 4		
	Intent	*Plan*
Mean	14.52650255	6.372533
Variance	58.70994825	20.8018
Observations	36	36
Pooled Variance	39.75587499	
Hypothesized Mean Difference	0	
Df	70	
t Stat	5.486617635	
P(T<=t) one-tail	3.0722E-07	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	6.14439E-07	
t Critical two-tail	1.994437112	

Table 0.22: Experiment Series 2. t-Test for intent recognition vs. plan recognition when observed agents were following plan 4.

t-Test: Two-Sample Assuming Equal Variances OA following plan 4		
	Norec	*Plan*
Mean	5.016577955	5.067088
Variance	37.14340462	24.94092
Observations	36	36
Pooled Variance	31.0421633	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-0.03846258	
P(T<=t) one-tail	0.484714175	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.96942835	
t Critical two-tail	1.994437112	

Table 0.23: Experiment Series 2. t-Test for plan recognition vs. no recognition when observed agents were following plan 4.

t-Test: Two-Sample Assuming Equal Variances OA following plan 4		
	Intent	*NoRec*
Mean	13.7656354	7.820526
Variance	64.73880852	30.58053
Observations	36	36
Pooled Variance	47.65966883	
Hypothesized Mean Difference	0	
Df	70	
t Stat	3.653596823	
P(T<=t) one-tail	0.000247818	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.000495635	
t Critical two-tail	1.994437112	

Table 0.24: Experiment Series 2. t-Test for intent recognition vs. no recognition when observed agents were following plan 4.

t-Test: Two-Sample Assuming Equal Variances OA following random plan		
	Intent	*Plan*
Mean	23.04668441	16.92265469
Variance	15.41047253	14.99638063
Observations	36	36
Pooled Variance	15.20342658	
Hypothesized Mean Difference	0	
Df	70	
t Stat	6.663506128	
P(T<=t) one-tail	2.54969E-09	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	5.09938E-09	
t Critical two-tail	1.994437112	

Table 0.25: Experiment Series 2. t-Test for intent recognition vs. plan recognition when observed agents were following a random plan.

t-Test: Two-Sample Assuming Equal Variances OA following random plan		
	Norec	*Plan*
Mean	24.76081171	16.74922
Variance	17.53301438	18.84955
Observations	36	36
Pooled Variance	18.19128218	
Hypothesized Mean Difference	0	
Df	70	
t Stat	7.969355512	
P(T<=t) one-tail	1.04017E-11	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	2.08034E-11	
t Critical two-tail	1.994437112	

Table 0.26: Experiment Series 2. t-Test for plan recognition vs. no recognition when observed agents were following a random plan.

Intent Recognition Vs No Recognition

t-Test: Two-Sample Assuming Equal Variances OA following random plan		
	Intent	*NoRec*
Mean	20.57889776	21.28
Variance	12.76830212	15.07744
Observations	36	36
Pooled Variance	13.92287283	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-0.797169055	
P(T<=t) one-tail	0.214024286	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.428048573	
t Critical two-tail	1.994437112	

Table 0.27: Experiment Series 2. t-Test for intent recognition vs. no recognition when observed agents were following a random plan.

Intent Recognition Vs Plan Recognition

t-Test: Two-Sample Assuming Equal Variances T: 50 Intent Vs Plan		
	Intent	*Plan*
Mean	19.81370592	14.77988
Variance	45.79140863	51.24576
Observations	54	54
Pooled Variance	48.51858573	
Hypothesized Mean Difference	0	
Df	106	
t Stat	3.755135434	
P(T<=t) one-tail	0.000141692	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	0.000283384	
t Critical two-tail	1.982597262	

Table 0.28: Experiment Series 2. t-Test for intent recognition vs. plan recognition. Time to communication 50.

t-Test: Two-Sample Assuming Equal Variances T: 50 Plan Vs No Rec		
	Norec	*Plan*
Mean	19.76968286	12.15103
Variance	137.6092332	53.13399
Observations	54	54
Pooled Variance	95.3716121	
Hypothesized Mean Difference	0	
Df	106	
t Stat	4.053689063	
P(T<=t) one-tail	4.82063E-05	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	9.64127E-05	
t Critical two-tail	1.982597262	

Table 0.29: Experiment Series 2. t-Test for plan recognition vs. no recognition. Time to communication 50.

t-Test: Two-Sample Assuming Equal Variances T: 50 Intent Vs No Rec		
	Intent	*NoRec*
Mean	16.05906705	18.25704
Variance	34.96588949	81.67175
Observations	54	54
Pooled Variance	58.31881961	
Hypothesized Mean Difference	0	
Df	106	
t Stat	-1.495547302	
P(T<=t) one-tail	0.06887151	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	0.137743021	
t Critical two-tail	1.982597262	

Table 0.30: Experiment Series 2. t-Test for intent recognition vs no recognition. Time to communication 50.

t-Test: Two-Sample Assuming Equal Variances T: 200 Intent Vs Plan		
	Intent	*Plan*
Mean	21.64522806	12.86715
Variance	58.89178585	42.82866
Observations	54	54
Pooled Variance	50.86022516	
Hypothesized Mean Difference	0	
Df	106	
t Stat	6.395757001	
P(T<=t) one-tail	2.2037E-09	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	4.4074E-09	
t Critical two-tail	1.982597262	

Table 0.31: Experiment Series 2. t-Test for intent recognition vs. plan recognition. Time to communication 200.

t-Test: Two-Sample Assuming Equal Variances T: 200 No Rec Vs Plan		
	Norec	*Plan*
Mean	19.69753086	10.70825
Variance	157.5003405	33.99938
Observations	54	54
Pooled Variance	95.74985866	
Hypothesized Mean Difference	0	
Df	106	
t Stat	4.773509082	
P(T<=t) one-tail	2.91327E-06	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	5.82655E-06	
t Critical two-tail	1.982597262	

Table 0.32: Experiment Series 2. t-Test for no recognition vs. plan recognition. Time to communication 200.

t-Test: Two-Sample Assuming Equal Variances T: 200 Intent Vs No Rec		
	Intent	*NoRec*
Mean	17.2058106	17.43195
Variance	47.03470003	66.77663
Observations	54	54
Pooled Variance	56.90566392	
Hypothesized Mean Difference	0	
Df	106	
t Stat	- 0.155769391	
P(T<=t) one-tail	0.438255529	
t Critical one-tail	1.659356034	
P(T<=t) two-tail	0.876511058	
t Critical two-tail	1.982597262	

Table 0.33: Experiment Series 2. t-Test for intent recognition Vs. no recognition. Time to Communication 200.

t-Test: Two-Sample Assuming Equal Variances One additional agent - all agent types		
	More	*Less*
Mean	15.40379012	15.10669
Variance	76.71365863	76.58749
Observations	108	108
Pooled Variance	76.6505746	
Hypothesized Mean Difference	0	
Df	214	
t Stat	0.249364546	
P(T<=t) one-tail	0.401658995	
t Critical one-tail	1.652005156	
P(T<=t) two-tail	0.80331799	
t Critical two-tail	1.971111258	

Table 0.34: Experiment Series 2. t-Test for one additional agent across all additional agent types.

t-Test: Two-Sample Assuming Equal Variances One additional agent - intent agent		
	More	*Less*
Mean	16.99962717	14.22117
Variance	83.63357647	51.50899
Observations	36	36
Pooled Variance	67.57128541	
Hypothesized Mean Difference	0	
Df	70	
t Stat	1.434031223	
P(T<=t) one-tail	0.078007767	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.156015535	
t Critical two-tail	1.994437112	

Table 0.35: Experiment Series 2. t-Test for one additional intent recognition agent.

t-Test: Two-Sample Assuming Equal Variances One additional agent - plan agent		
	More	*Less*
Mean	13.47177866	16.71939
Variance	72.18996672	96.78019
Observations	36	36
Pooled Variance	84.48507883	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-1.499031993	
P(T<=t) one-tail	0.069180958	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.138361916	
t Critical two-tail	1.994437112	

Table 0.36: Experiment Series 2. t-Test for one additional plan recognition agent.

t-Test: Two-Sample Assuming Equal Variances One additional agent - non recognition agent		
	More	*Less*
Mean	15.73996452	14.37952
Variance	72.12605359	81.82416
Observations	36	36
Pooled Variance	76.97510514	
Hypothesized Mean Difference	0	
Df	70	
t Stat	0.657873425	
P(T<=t) one-tail	0.25638859	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.512777179	
t Critical two-tail	1.994437112	

Table 0.37: Experiment Series 2. t-Test for one additional non-recognition agent.

t-Test: Two-Sample Assuming Equal Variances 5 additional agents - all agent types		
	More	*Less*
Mean	16.52720485	14.01715
Variance	98.50320238	70.60035
Observations	108	108
Pooled Variance	84.55177686	
Hypothesized Mean Difference	0	
Df	214	
t Stat	2.005942914	
P(T<=t) one-tail	0.023060682	
t Critical one-tail	1.652005156	
P(T<=t) two-tail	0.046121365	
t Critical two-tail	1.971111258	

Table 0.38: Experiment Series 2. t-Test for five additional agents across all additional agent types.

t-Test: Two-Sample Assuming Equal Variances 5 additional agents - intent agents		
	More	*Less*
Mean	19.48702595	12.98935
Variance	77.45687889	58.43009
Observations	36	36
Pooled Variance	67.94348287	
Hypothesized Mean Difference	0	
Df	70	
t Stat	3.344414434	
P(T<=t) one-tail	0.000663846	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.001327692	
t Critical two-tail	1.994437112	

Table 0.39: Experiment Series 2. t-Test for five additional intent recognition agents.

t-Test: Two-Sample Assuming Equal Variances 5 additional agents - plan agents		
	More	*Less*
Mean	13.02029275	15.83156
Variance	78.04865126	95.05123
Observations	36	36
Pooled Variance	86.54994121	
Hypothesized Mean Difference	0	
Df	70	
t Stat	-1.282049694	
P(T<=t) one-tail	0.10202754	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.204055081	
t Critical two-tail	1.994437112	

Table 0.40: Experiment Series 2. t-Test for five additional plan recognition agents.

t-Test: Two-Sample Assuming Equal Variances 5 additional agents - non recognition agents		
	More	*Less*
Mean	17.07429585	13.23054
Variance	123.6643128	57.24492
Observations	36	36
Pooled Variance	90.45461693	
Hypothesized Mean Difference	0	
Df	70	
t Stat	1.714655194	
P(T<=t) one-tail	0.045416453	
t Critical one-tail	1.666914479	
P(T<=t) two-tail	0.090832906	
t Critical two-tail	1.994437112	

Table 0.41: Experiment Series 2. t-Test for five additional non-recognition agents.

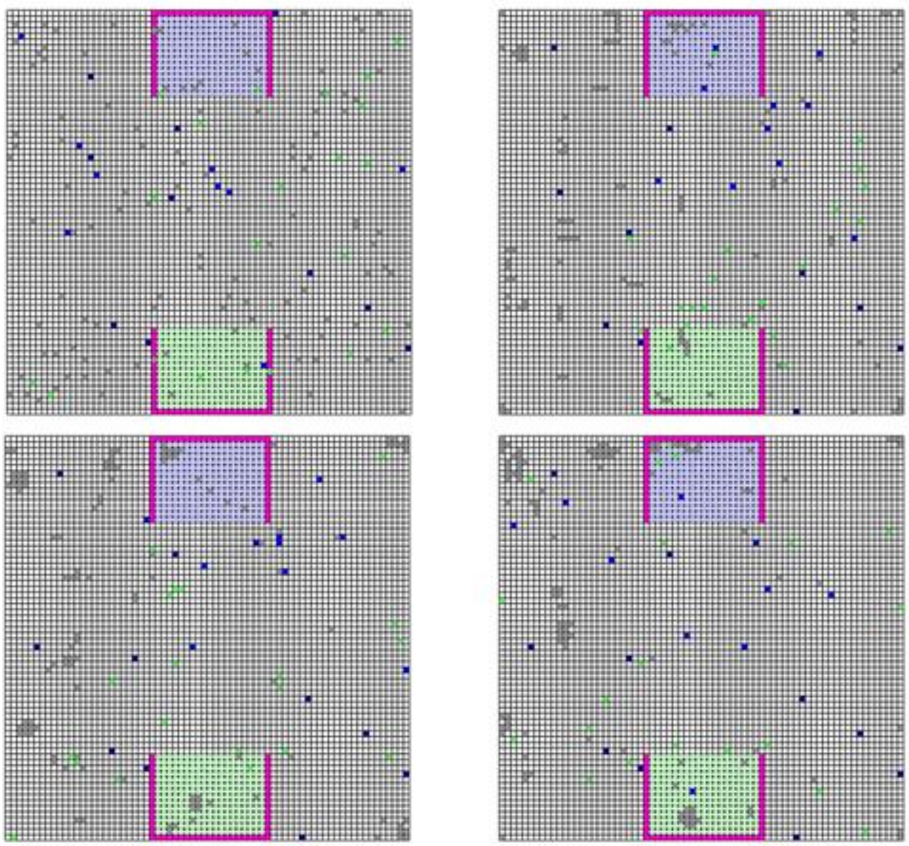


Figure 0.1: Screen captures from cow herding simulation. Shows timesteps 0, 25, 50, and 100. All agents begin in a random location.

References

Agah, A. (1995). “Robot Teams, Human Workgroups and Animal Sociobiology: A Review of Research on Natural and Artificial Multi-Agent Autonomous Systems,” *Advanced Robotics Journal*, 10: Number 6, 523-545.

Ahmad, N. and A. Agah. (2013). “Plan and intent recognition in a multi-agent system for collective box pushing.” *Journal of Intelligent Systems*, DOI: 10.1515/jisys-2013-0044.

Ahmad, N. and A. Agah (in review). “Intent Recognition in Multi-Agent Systems: Cow Herding.” *International Journal of Pattern Recognition and Artificial Intelligence.*

Avrahami-Zilberbrand, D., G. Kaminka and H, Zarosim (2005). “Fast and Complete Symbolic Plan Recognition: Allowing for Duration, Interleaved Execution, and Lossy Observations,” *In Proceedings of the AAAI Workshop on Modeling Others from Observations Workshop* (MOO-05), Edinburgh, Scotland.

Avrahami-Zilberbrand, D. and G.A. Kaminka (2006). “Hybrid Symbolic-Probabilistic Plan Recognizer: Initial steps.” *In Proceedings of the AAAI Workshop on Modeling Others from Observations (MOO-06),* Boston, MA.

Banerjee, B., L. Kraemer, and J. Lyle (2010). “Multi-Agent Plan Recognition: Formalization and Algorithms,” *In Proceedings of the 24th AAAI Conference on Artificial Intelligence* (AAAI-10). Atlanta, Georgia, 1059-1064.

Bellifemine, F., A. Poggi, and G. Rimassa (2001). "Developing multi-agent systems with a FIPA-compliant agent framework," *Software-Practice and Experience*, 31, 103-128.

Bigelow, D. C. (2013). “Intent Recognition in Multi-Agent Domains” (Master of Science Thesis, University of Nevada).

Boss, N. S., A. S. Jensen, and J. Villadsen. (2009). Developing artificial herders using Jason. arXiv preprint arXiv:1001.0115.

Briggs, W. and D. Cook. (1995). “Local planning and teamwork: Minimizing Communication in Multi-Agent Planning”, *Technical Report Number TR-CSE-95-001*, *Department of Computer Science and Engineering*, University of Texas at Arlington, Arlington, TX, 688-693.

Castelfranchi, C. (2006). “SILENT AGENTS: From Observation to Tacit Communication,” *Ibero-American Conference on AI* (*IBERAMIA-SBIA 2006)*, Ribeirão Preto, SP, Brazil, 98-107.

Dechter, R., and J. Pearl (1985). "Generalized best-first search strategies and the optimality of A*" *Journal of the ACM*, 32: 505-536.

Dix, J., M. Köster, and F. Schlesinger. (2008). “Agent Contest 2008.” *Multi-Agent Programming Contest.* Retrieved July 16, 2013, from http://www.multiagentcontest.org/2008.

Dix, J., M. Köster, and F. Schlesinger. (2010). “Agent Contest 2010.” *Multi-Agent Programming Contest.* Retrieved July 16, 2013, from http://www.multiagentcontest.org/2010.

Dix, J., M. Köster, and F. Schlesinger. (2013). “Multi-Agent Programming Contest.” *Multi-Agent Programming Contest*. Retrieved July 30, 2013, from http://www.multiagentcontest.org.

Eclipse [Online] (2012). Retrieved July 16, 2013, from http://www.eclipse.org/.

Gabaldon, A. (2009). "Activity Recognition with Intended Actions." *In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, Pasadena, California, 1696-1701.

Gal, Y. and A. Pfeffer (2003). "A Language for Opponent Modeling in Repeated Games." *Autonomous Agents and Multiagent Systems* (*AAMAS03) Game theoretic workshop*, Melbourne, Australia, 265-272.

Geib, C. and S. Harp, (2004). "Empirical analysis of a probabilistic task tracking algorithm" *In Proceedings of Modeling Other Agents from Observations (MOO 2004) Workshop*, New York, NY, 65-71.

Goldman, RP., F. Kabanza, and P. Bellefeuille (2010). "Plan Libraries for Plan Recognition: Do We Really Know What They Model?" *In Plan, Activity, and Intent Recognition (PAIR) 2010, AAAI Workshop Proceedings*, AAAI, Atlanta, Georgia.

Hackman, J. R. (1990). *Groups That Work (and Those That Don't).* Jossey-Bass Inc., Publishers, San Francisco, California.

Heßler, A., Hirsch, B., and T. Küster. (2010). "Herding cows with JIAC V." Annals of Mathematics and Artificial Intelligence, 59(3-4): 335-349.

Hongeng, S. and R. Nevatia (2001). "Multi-agent event recognition", *In Proceedings of IEEE International Conference on Computer Vision*, Vancouver, Canada, 84-91.

Huang, Z., Y. Yang, and X. Chen (2003). “An approach to plan recognition and retrieval for multi-agent systems.” *In Workshop on Adaptability in Multi-Agent Systems, First RoboCup Australian Open 2003 (AORC-2003)*, Sydney, Australia.

Huber, M.J. and E.H. Durfee, (1995). “On acting together: Without communication”, *In Working Notes of the AAAI Spring Symposium on Representing Mental States and Mechanisms,* Stanford, CA, 60-71.

Huber, M.J. and E.H. Durfee, (1995). “Deciding when to commit to action during observation-based coordination” *In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* AAAI Press, San Francisco, CA, 163-170.

IEEE Foundation for Intelligent Physical Agents. (2012). *Foundation for Intelligent Physical Agents.* Retrieved July 16, 2013, from http://www.fipa.org/.

Intille, S. S. and A. F. Bobick (1999). “A framework for recognizing multi-agent action from visual evidence.” *In Association for the Advancement of Artificial Intelligence / Innovative Applications of Artificial Intelligence (AAAI/IAAI'99)*, Orlando, Florida, 518-525.

Jordan, P.W. and M.A. Walker (1996). “Deciding to remind during collaborative problem solving: Empirical evidence for agent strategies” *In Proceedings of the 14th Conference of the American Association of Artificial Intelligence*, Portland, Oregon, AAAI Press, 16-24.

Kanevsky, D., J. Reisinger, R. Sicconi, and M. Viswanathan (2013). "Hierarchical methods and apparatus for extracting user intent from spoken utterances," US Patent 20130006637.

Kitano, H., M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa (1997). "RoboCup: The Robot World Cup Initiative," *Proceedings of the first international conference on Autonomous agents*, 340-347

Komenda, A., P. Novák, and M. Pechou (2013). "How to Repair Multi-agent Plans: Experimental Approach," *In Proceedings of Distributed and Multi-agent Planning (DMAP) Workshop of 23rd International Conference on Automated Planning and Scheduling* (ICAPS'13).

Lin, C. and Y. Hsu (2006). "IPARS: Intelligent Portable Activity Recognition System via Everyday Objects, Human Movements, and Activity Duration." *Modeling Others from Observations, Boston, MA (MOO 2006)*, Technical Report WS-06-13, 44-52.

Majchrzak, A. and Gasser, L. (1992). "Toward a Conceptual Framework for Specifying Manufacturing Workgroups Congruent with Technological Change." *International Journal of Computer Integrated Manufacturing*, 5: 118-131.

Mammen, D., and Lesser, V. (1997). "A Test Bed for the Evaluation of Multiagent Communication and Problem-Solving Strategies." *In Constraints and Agents: Papers from the 1997 AAAI Workshop,* Menlo Park, Calif.: AAAI Press. Technical Report WS-97-05, 32-38.

Marhasev, E., M. Hadad, and G.A. Kaminka. (2006). "Non-stationary Hidden Semi Markov Models in Activity Recognition." *In Proceedings of the AAAI Workshop on Modeling Others from Observations*, Boston, MA (MOO-06), 53-60.

Mataric, M. J. (1995). "Issues and Approaches in the Design of Collective Autonomous Agents," *Journal of Robotics and Autonomous Systems*. 16: 321-331.

McEleney, B., and G. O'Hare (2004). “Planning to Replan in a Multi-agent Environment,” *In Proceedings of the 23rd Workshop of the UK Planning and Scheduling Special Interest Group* December 20 - 21st 2004, Cork, Ireland.

McEleney, B., and G. O'Hare (2004). “A Planner for Plan Construction Plans for Multi-Agent Plans,” *In Proceedings of 15th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'04)* September 8th - 10th 2004, Castlebar, Ireland.

McEleney, B., and G. O'Hare (2005). “Efficient Dialogue Using a Probabilistic Nested User Model.” *In Proceedings of the Fourth IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems* August 1st 2005, Edinburgh, Scotland.

Molineaux, M., D. Aha, and G. Sukthankar (2009). “Beating the Defense: Using Plan Recognition to Inform Learning Agents,” *In Proceedings of 22nd Florida Artificial Intelligence Research Society (FLAIRS),* Sanibel Island, Florida, 257-262.

Rahmani, A. T., A. Saberi, M. Mohammadi, A. Nikanjam, E.A. Mosabbeb, and M. Abdoos. (2009). SHABaN Multi-agent Team To Herd Cows. *In Programming Multi-Agent Systems,* 5442: 248-252. Springer LNCS, Berlin Heidelberg.

Repast [Online] (2012). Retrieved July 16, 2013, from http://repast.sourceforge.net/ .

Riedl, M.O., and R. Young (2004). “An Intent-Driven Planner for Multi-Agent Story Generation.” *In AAMAS '04 Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems,* New York, New York, 1: 186-193.

Rish, I. (2001). “An Empirical Study of the Naive Bayes Classifier.” *In Proceedings of IJCAI-01 workshop on Empirical Methods in AI, International Joint Conference on Artificial Intelligence 2001*, Seattle, Washington, 41-46.

Rodriguez, S., V. Hilaire, N. Gaud, S. Galland, and A. Koukam (2011). "Holonic Multi-Agent Systems." *Self-organising Software: Natural Computing Series 2011*, Springer, New York, New York, 251-279.

Sadilek, A. and H. Kautz (2010). “Modeling and reasoning about success, failure, and intent of multi-agent activities.” In Mobile Context-Awareness Workshop, Twelfth ACM International Conference on Ubiquitous Computing.

Shoham Y. and K. Leyton-Brown (2009). “Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.” Cambridge University Press, Cambridge, UK.

Su H., X. Wang, and Z. Lin (2009). “Flocking of Multi-Agents With a Virtual Leader.” *IEEE Transactions on Automatic Control*, 54: Issue 2, 293-307.

Tabachnick, B. G., Fidell, L. S., and Osterlind, S. J. (2001). “Using multivariate statistics.” Boston: Pearson.

Tian, Y. and Y. Zhang (2012). "High-order consensus of heterogeneous multi-agent systems with unknown communication delays," *Automatica*, Volume 48, Issue 6, June 2012, 1205-1212.

Tsatsoulis, C., N. Ahmad, E. Komp, and C. Redford (2008). “Using a Contract Net to Dynamically Configure Sensor Webs,” *IEEE Aerospace Conference*, Big Sky, MT, 1-6.

Walker, M. A. and O. Rambow (1994). “The role of cognitive modeling in achieving communicative intentions” *In Proceedings of The 7th International Conference on Natural Language Generation*, New Forest, England, 171-180.

Wen, G., Duan, Z., Yu, W., and Chen, G. (2012). “Consensus in multi-agent systems with communication restraints,” *International Journal of Robust and Nonlinear Control*, 170-182.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, UK.

Wu, Q., J. Zhou, and L. Xiang (2012). "Impulsive consensus seeking in directed networks of multi-agent systems with communication time delays," *International Journal of Systems Science,* 43: Issue 8, 1479-1491

Xuan, P. and V. Lesser and S. Zilberstein (2000). “Formal Modeling of Communication Decisions in Cooperative Multi-agent Systems”, *In Proceedings of the Second Workshop on Game Theoretic and Decision Theoretic Agents (GTDT 2000)*, Boston, MA.

Xuan, P. , V. Lesser, and S. Zilberstein (2001). “Communication decisions in multi-agent cooperation: Model and experiments.” *In Proceedings of the Fifth International Conference on Autonomous Agents*, Montreal, Canada, 616-623.

Ya’akov, G., S. Reddy, S. Shieber, A. Rubin, and B. Grosz (2012). “Plan recognition in exploratory domains.” *Artificial Intelligence*. 176: Issue 1, 2270-2290.

Zhang X., H. Huang, and Q. Yang (2012). "Implementing an FPGA system for real-time intent recognition for prosthetic legs," *49th Design Automation Conference (DAC),* San Francisco, 169-175.