Evolutionary Learning of Goal-Driven Multi-Agent Communication

By

©2016
Alhanoof Althnian

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

Committee members

Dr. Arvin Agah, Chairperson

Dr. Prasad Kulkarni

Dr. Bo Luo

Dr. Fengjun Li

Dr. Elaina Sutley

Date defended: September 6th, 2016

The Dissertation Committee for Alhanoof Althnian certifies
that this is the approved version of the following dissertation :

Evolutionary Learning of Goal-Driven Multi-Agent Communication

Dr. Arvin Agah, Chairperson

Date approved:

Abstract

MULTI-AGENT systems are a common paradigm for building distributed systems in different domains such as networking, health care, swarm sensing, robotics, and transportation. Systems are usually designed or adjusted in order to reflect the performance trade-offs made according to the characteristics of the mission requirement.

Research has acknowledged the crucial role that communication plays in solving many performance problems. Conversely, research efforts that address communication decisions are usually designed and evaluated with respect to a single predetermined performance goal. This work introduces Goal-Driven Communication, where communication in a multi-agent system is determined according to flexible performance goals.

This work proposes an evolutionary approach that, given a performance goal, produces a communication strategy that can improve a multi-agent system's performance with respect to the desired goal. The evolved strategy determines what, when, and to whom the agents communicate. The proposed approach further enables tuning the trade-off between the performance goal and communication cost, to produce a strategy that achieves a good balance between the two objectives, according the system designer's needs.

Acknowledgements

ALL PRAISE is due to God, the Lord of the Worlds, for granting me the strength and knowledge to understand, learn, and complete this dissertation, and the ultimate peace and blessings be upon His Prophet Muhammad and upon all His Prophets and Messengers.

I am deeply thankful to my beloved parents, Hessah and Abdallah, for their prayers, unconditional love, and endless support. You are literally the greatest gifts in my life and the best thing ever happened to me.

Heartfelt thanks go to my soulmate and partner in this journey, Dr. Yasser. Thank you for being next to me, celebrating the ups and strengthening me during the downs of this journey, and thank you for making our life such an exciting adventure. Very special thanks to my little Mazen for bringing so much happiness to my life, for his patience, and for forgiving me about all the bedtime stories I missed. My incredible sisters and brothers, thank you for believing in me and for your encouragement. I love you all.

I would like to thank and express my sincere gratitude to the best advisor that any doctoral researcher can wish for, Dr. Arvin Agah. Your encouragement at our first meeting and throughout the years has been the spark that pushed me forward and kept me motivated. Thank you for your valuable guidance and feedback, continuous support, and great kindness. I am really indebted for your flexibility and for accommodating my remote work schedule. Thank you!

I would like also to thank Dr. Prasad Kulkarni, Dr. Bo Luo, Dr. Fengjun Li, Dr.

Elaina Sutley, Dr. Jerzy Grzymala-Busse, and Dr. Sarah Kieweg, for serving on my committee and for their discussion, questions, and feedback.

I am really grateful to the staff of Advanced Computing Facility at ITTC, especially Paul Calnon and Wesley Mason. Thank you for your patience, prompt responses, unparalleled service, and for helping me meet my deadlines. Without your assistance, my research would have been much harder.

I am also thankful to all my friends, including those I made during this journey and everywhere I lived, in Florida, Indiana, Kansas, Arizona, and New York. Your love and encouragement and our memories made a remarkable difference in this journey.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

THE PRIMARY goal of research in many computing systems is to improve performance (Kinney & Tsatsoulis, 1998). In multi-agent systems (MAS), however, performance can be measured with respect to multiple, conflicting metrics, which can be viewed as different dimensions of the system's performance. The consideration and importance of each metric depend merely on the characteristics of the application domain and preferences of the system's designer.

Multi-agent systems have been commonly used for modeling and solving distributed and complex problems in different domains such as networking, health care, swarm sensing, robotics, transportation, and military. These domains vary in their specifications, requirements, and costs, and consequently in their performance goals. For example, some application domains are time-critical such as robot rescuers, where minimizing time has the first performance priority. In such domains the system designer might be willing to increase communication and energy consumption if it can help save more lives (Preist & Pearson, 1998). However, other applications are energy-critical, such as space/undersea operations, where energy is limited, and hence agents are required to complete their assigned task with the minimum possible amount of energy (Balch & Arkin, 1994). Therefore, the choice of a performance goal in MAS is a crucial decision to make (Kinney & Tsatsoulis, 1998; Balch & Arkin, 1994).

A powerful tool to customize performance is communication, which facilitates coordination of agents' actions. A number of research efforts have investigated the importance of communication and its impact on the performance of multi-agent systems. Studies are usually conducted by varying the communication conditions and testing the performance of the system. The work in (Balch & Arkin, 1994) and (Wei et al., 2014), carried out experiments to study the effect of communicating different types of information when agents are assigned different tasks. As shown in Figure 1.1a, the process starts by manually determining the type of information that agents are allowed to communicate, namely, none, only goals, only beliefs, or goals and beliefs. Then, the average performance of the system over multiple runs is calculated with respect to different metrics such as time to complete, interference, communication efficiency, and duplication of efforts. In their work, whenever agents are allowed to communicate information, they broadcast every value update once obtained. Results suggested that varying the type of information that agents communicate can significantly affect the performance of the multi-agent system with respect to different metrics, especially if no implicit communication is present. Moreover, their results showed that more communication does not always guarantee better performance, which has a crucial implication. It indicates that even in applications where communication is free, system designers should not allow full communication and assume that the system is performing at its best level.

Another related work, (Dowell & Stephens, 2001), investigated the effect of changes in coordination and communication parameters on the multi-agent performance with respect to time, number of moves, and the weighted linear combination of the average time and maximum time for three teams involved in multiple team pursuit problem. The authors' goal was to manually determine improved values for the parameters by running experiments with variant parameters' values. Three parameters were considered, one communication parameter, the n-th move value, representing the number of moves a team must wait before trying to communicate, and two coordination parameters, namely time-out counter, and time-distance threshold. Improved values for the parameters were identified, yet they only outperformed the other values considered, and hence cannot be considered the best values nor the result performance as the best performance. Another work by (Rybski et al., 2007) compared the time that it took a group of robots to perform a forage task with three implicit communication strategies, namely, no communication, reflexive communication (state communication), and deliberative communication (target location communication). Moreover, (Hurt, 2005) compared the performance of two bee-inspired communication strategies, namely, hives communication and site communication, and (Conforth & Meng, 2008) varied the amount of communication (none, low, or high) and compared the communication benefits.

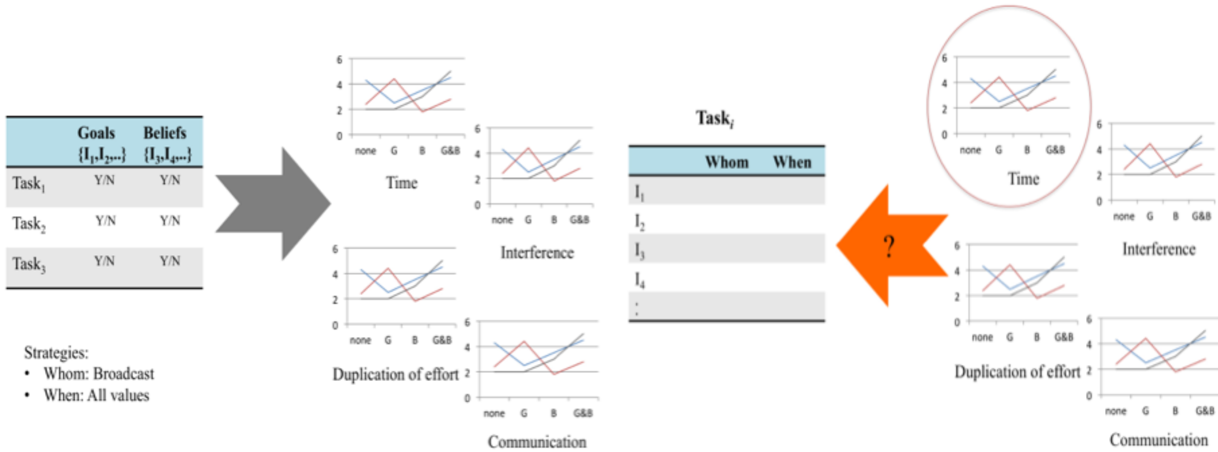
State of the art research has made progress in understanding the impact of communication variation on MAS performance, as well as emphasized the fact that system designers must choose a performance goal. However, results obtained are only consequences of manual communication modification, which open the door to many questions. For further explanation, we assume that the system has the best performance with respect to time when agents are allowed to communicate their beliefs. The following questions can be asked:

- Is it really necessary to communicate all agents beliefs in order to achieve the reported time performance?
- Are the communicated beliefs the best set of information instances that can be communicated to improve the system's performance with respect to time?
- Is it really necessary to broadcast all value updates of the communicated information instances to achieve the reported good performance?
- Is this the best performance that the system can achieve with respect to time?

The first question examines the contribution of the communicated information instances in minimizing the time that agents take to complete the assigned task. We argue that, for some cases, it is likely that only a subset of the communicated information had considerably contributed in helping agents finish the task in shorter time. This takes us to the second question, which suggests that since more communication does not guarantee better performance, it might be the case that restricting communication to only a subset of the beliefs can further decrease the time to complete the task. Likewise, combining a subset of the beliefs with a subset of the goals can probably result in further improved performance. The third question concerns the communication efficiency, as it argues that the same good performance can possibly be achieved with less communication by combining values into one message or communicating only important value updates to recipients who can make use of them.

The fourth question is specifically inspiring to us. It implies that better performance that results from manually designed communication decisions does not necessarily represent the best performance of the system. Going back to the point where we emphasized the fact that system designers must decide on their performance goal based on the domain's characteristics, it would be extremely useful if the process presented in Figure 1.a can be reversed. Therefore, rather than manually creating different communication conditions, the system designer can start from selecting the performance goal for one task, feed it to some learning system, and then results determine what, when, and to whom information instances should be communicated in order to achieve the best performance of the system with respect to the selected goal (Figure 1.1b).

The prominence of this approach comes from the fact that information instances, which can be obtained by agents during task execution, are of varying quality, and each information instance has changing effects on the performance with respect to different metrics. Besides, communication is usually costly, and system designers have specific performance preferences according to the domain's requirements. This lends itself well to the concept of goal-driven communication. Therefore, depending on the performance goal, the communication strategy should address the question of whether each information instance should be communicated, and if so, the strategy should determine when and to whom it should be communicated.



(a) Investigating the impact of varying communication conditions on MAS performance, e.g. (Balch & Arkin, 1994; Wei et al., 2014).

(b) Learning a goal-driven communication strategy for improving MAS performance with respect to a user-defined goal.

Figure 1.1: Reversal of the (a) communication-to-performance investigation process to obtain (b) performance-to-communication learning process.

1.2 Problem Statement

Given a set of k information instances, a performance goal P, and performance goal's weight α, determine what, when, and to whom information instances should be communicated among agents that achieve reasonable system performance with respect to P, and a good trade-off, according to α, between the communication cost and performance with respect to P.

1.3 Research Hypothesis

We hypothesize that communication strategies in multi-agent systems, namely what, when, and to whom agents communicate, can be learned in order to improve the performance of the system with respect to flexible performance goals, resulting in learning of goal-driven communication strategies.

In order to test the research hypothesis, we will utilize Genetic Algorithms to learn goal-driven communication strategies. It is expected that the evolved strategies will vary according to the performance goals, and the performance of the multi-agent system will improve when using the corresponding communication strategies.

1.4 Contributions

The work in this dissertation draws on scholarly work that examine the impact of communication variation on performance, and will contribute to addressing multi-agent communication decisions as well as performance improvements. The practical contributions of proposing, designing, and evaluating an evolutionary learning approach for goal-driven multi-agent communication are five-fold:

1. The learned communication strategy guides agents on all their communication decisions.

2. The proposed approach allows system designers to easily vary the goal and automatically obtain the corresponding communication strategy. Therefore, the system designer does not need to know or analyze the properties of each information instance and its effect on the performance goal of the system, which can eliminate a significant design task in developing a multi-agent system.

3. The approach provides a tool for customizing the tradeoff between the system's performance and communication cost.

4. The proposed approach can assist system designers to figure out the potentially best performance that the system can achieve with respect to a specific goal, such as the minimum time or energy that a task takes to complete. Therefore, a system designer will be able to choose among the performance of the system with multiple communication strategies of varying goals and select the one that has the best fit to the system's needs.

5. Analysis of the system's performance with respect to different performance metrics while

executing communication strategies of varying goals and under different conditions can provide insights into how the different performance metrics are related to each other.

The theoretical contributions of this work are as follows:

1. We demonstrate that what, when, and to whom agents communicate, and consequently their performance, can vary significantly when different goals or parameter settings are applied.
2. We show under which conditions a communication strategy with goal P_1 may outperform another with goal P_2, with respect to P_2.
3. We show under which conditions increasing performance goal's weight does not result in better performance with respect to the desired goal.
4. We classify information instances into three categories according to their influence on a performance goal, namely favorable, neutral, and unfavorable information instances.

We will revisit, later in this dissertation, some of the contributions in each category to explain in more details how they have been accomplished.

1.5 Dissertation Structure

This dissertation is organized into seven chapters:

- Chapter 2 provides background information on multi-agent systems and Genetic Algorithms (GA), and presents an overview of related work.
- Chapter 3 defines communication strategy, and its components, as well as analyzes the problem of designing a goal-driven communication strategy for a multi-agent system, and presents how we utilize a genetic algorithm to design a learning system that automatically generates an effective goal-driven communication strategy.

- Chapter 4 introduces our two case studies, and discusses the research methodology that we adopt to conduct this research. It also explains parallelization of GA.
- Chapter 5 provides details on the Wumpus World, the first multi-agent case study, and presents the results we obtained for this domain along with analysis.
- Chapter 6 presents the second multi-agent case study, Collective Construction, as well as the results obtained from the application of the proposed approach.
- Chapter 7 provides concluding remarks. We further describe the contributions, limitations, and future work.

CHAPTER 2

Background and Related Work

2.1 Multi-Agent Systems (MAS)

Whenever a new concept emerges in Computer Science, efforts are directed towards defining it, and the field of Multi-Agent Systems (MAS) is not an exception. Multi-agent systems are classified as distributed systems. Yet, characteristics of the computational entity, i.e., agent, that constitutes multi-agent systems are what distinguish the two paradigms. One of the popular definitions of MAS is that of (Barbuceanu & Fox, 1996), which states “a Multi-Agent System is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver" . The main components of a MAS are agents and an environment.

Environments are characterized by their observability, determinism, episodism, stasis, and discreteness (Russell & Norvig, 2003). Variations in these characteristics result in different types of environments, which is the main factor in deciding on a solution method. If an agent is operating in a fully observable environment, it can observe the complete state (has a global view) of the environment at any point in time. A deterministic environment (versus stochastic) is completely predictable since the state is determined entirely by the current state and agents' actions. Further, episodic environment divides the problem into independent episodes, where an episode consists of an agent sensing the environment followed by its action. When the environment's states are finite, we say that the environment is discrete. Yet, we say that an environment is static if it remains unchanged while an agent is acting.

Agents are situated in the environment, which they perceive and interact with. Several definitions of an agent have been proposed in the past, such as(Russell, 1997; Shoham, 1993; Franklin & Graesser, 1996). However, the main agent's properties are included in the definition by (Wooldridge & Jennings, 1994), which suggests that an agent is "a hardware or (more usually) a software-based computer system that enjoys the following properties: autonomy - agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state; social ability - agents interact with other agents (and possibly humans) via some kind of agent-communication language; reactivity: agents perceive their environment and respond in a timely fashion to changes that occur in it; pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking initiative". At any point in time, an agent's decision and action (output) depend on several inputs including prior knowledge of the environment and other agents, observations, and goals. When communication is allowed, information received from others are also used to make decisions. Agents can be divided into many types with variant complexity, depending on how they perceive and interact with the environment (Russell & Norvig, 2003). The first and simplest type, known as Simple Reflex Agent, operates successfully in only fully observable environments. The behavior of this type of agent is based on condition-action rules, where conditions are the current percept. When the percept's history is considered, agents need to maintain an internal model of the world, hence the Model-based Reflex Agents. Unlike simple agents, model-based agents can operate successfully in partially observable environments. The third type of agents are further enhanced by including goal information. Goal-based agents are aware of the goal states, which allow them to evaluate possible decisions and choose the one that yields a goal state. Utility-based agents can further compute a utility value of each state, which measures how happy the agent would be in the corresponding state, and hence its goal is to maximize its happiness. Learning Agents can operate in unknown environment and improve their performance using feedback for behavior.

2.2 Communication in MAS

Empowering artificial agents with the ability to communicate with each other involves tackling multiple problems. For instance, similar to humans, communicating agents need a mutual language to be able to understand each other. Agents also need the ability to decide when they should talk, to whom, and what to say, as communication incurs cost. Research efforts exist in the literature that focus on solving these communication problems. In this section, we shed light on the main communication challenges in multi-agent systems; namely, timing, selection, language, protocols, interpretation, and ambiguity, and we discuss the corresponding approaches that have been proposed in the literature for solving each problem.

2.2.1 Communication Timing

At each time-step, an agent makes a decision as to whether to communicate or not. This decision is dependent on the benefit of communicating the available information at the current time-step to the overall performance. Yet, this communication decision is critical because over time, information may lose its value. For example, sending information at the current time-step could save a team member from re-deriving the same information, or it could correct old no-longer-true information, while delaying the communication decision may result in the team member obtaining the information by itself. Factors to be considered for this decision are not only whether communication is beneficial, but also whether it is worthwhile to communicate, given the communication cost. The trade-off between communication benefit and cost is further discussed in a later section.

Multiple approaches have been proposed in the literature to guide agents on when to communicate. Some works let agents communicate only at fixed predetermined points in time (Speranzon

& Johansson, 2003), or if a specific situation takes place. Examples of the latter include if history inconsistency occurs (Wu et al., 2009), when the agent's current plan cannot be achieved (Xuan et al., 2000), when their sub-goals are achieved (Goldman & Zilberstein, 2003), or when an agent has to backtrack, or after assigning a certain number of variables (Mammen & Lesser, 1997). In the work by (Xuan et al., 2001), all agents are enforced to synchronize their local states if at least one agent decides to communicate. A research effort by (Zhang, 2007) adopted a decision-theoretic approach to allow agents to reason online about their communication decision, based on two factors to value information: timeliness (using the most recent received but potentially outdated value), and relevance (waiting for the new value). In this approach, researchers designed multiple communication strategies for information provider agents as well as information needer agents to choose from. A provider agent can either ProactiveTell or Silence, for information obtained, or it can Reply, WaitUntilNext, or Reject, for each request received from others. A needer agent can ActiveAsk, Silence, or Wait, for information needed, or it can either Accept or RejectNeed received information.

One of the popular decision-theoretic models of multi-agent decision making is Dec-MDP, whose solution is intractable due to the high complexity (NEXP-complete) (Bernstein et al., 2002). Therefore, wide range of research efforts have focused on sub-classes of the problem that have less complexity, such as (Becker et al., 2004a) and (Becker et al., 2004b). Also, researchers usually base their work on one or more simplified assumptions to reduce computation. In the work by (Roth, 2007), an algorithm called *ACE_PJB_COMM* is proposed to provide agents with the heuristic to decide when to communicate. The algorithm is designed such that an agent should communicate only when the communicated information is likely to change the joint action of the team, which is accessible to all agents. Specifically, this joint action change has to increase the expected reward by more than the cost of the communication for the agent to communicate (Roth, 2007). Furthermore, (Mostafa & Lesser, 2009) focused on problems with structured interactions and introduced a new model, Event-Driven Interactions with Complex Rewards (EDI-CR) for offline communication planning. Their work proposed heuristics that analyze interactions among the agents' sub-problems and identify potential communication points. The proposed heuristics include adding communication after critical actions, after actions with very different outcomes, and lastly where it causes most belief change. The work by (Abdelmoumène & Belleili, 2015) exploits problem structure in order to limit communication decisions in theoretic decision models that operate under time pressure having uncertain actions durations. Some researchers borrow inspirations from other areas to design communication-timing heuristics. Examples include the work by (Reddy et al., 2012), which uses a game theory approach and (Dutta et al., 2007), which uses information redundancy to address when agents should communicate. The computation, however, is usually based on myopic assumptions, where each agent evaluates the benefit of communication in isolation for 1-step horizon assuming others never communicate.

Another direction of research is to let agents learn when to communicate. In the work of (Kinney & Tsatsoulis, 1998), agents learn the frequency with which information should be communicated to a neighbor agent using a system of equations that generates probability tables. Moreover, (Ghavamzadeh & Mahadevan, 2004) introduced Com-Cooperative Hierarchical Reinforcement Learning (HRL) to allow agents make rational communication timing decisions as well as action decisions. Using the HRL, the main task, to be completed by agents, is broken down by the system designer into subtasks; each with a set of termination states. Subtasks, whose performance can improve by agents cooperation, are called cooperation subtasks and placed at a higher level of the hierarchy, called the cooperative level. However, subtasks that need no cooperation are placed a lower level in the hierarchy. A communication level is added to the task hierarchy under each cooperation level, and a multi-agent policy is defined for cooperative tasks, while a single-agent policy is defined for non-cooperative tasks. The authors proposed an action-value function for executing a subtask in the context of a parent task when the agent is in a state s, which can be used to decide whether to communicate (perform the communication subtask) or not. The function sums the value of executing the sub-task and the value of completing the parent task afterward. The decision is made by comparing the value of the function with communication and without communication.

2.2.2 Communication Selection

Communication selection refers to selecting information to communicate, i.e., what to communicate, and selecting communication recipients, i.e., who to communicate with. During task execution, agents can obtain different types of information, such as observations, state, results of subtasks, and goals. Deciding which of the obtained information is essential to communicate and to whom it is essential is an important and challenging problem. It is important because if an agent is unable to make such communication decisions, then whenever it decides to communicate, it will broadcast its whole information history. The problem is also challenging because agents need information about their teammates' states in order to be able to make such decisions, which is, in practice, inaccessible.

One common approach to the selection problem is synchronization, which works as follows. If at least one agent decides to communicate, then all agents are forced to communicate their information with their teammates. The communicated information include only newly obtained information since the last communication. This approach has been adopted by many researchers (Xuan et al., 2001; Carlin & Zilberstein, 2009; Wu et al., 2009) because it has the advantage of ensuring that agents share the same belief about the world, whenever communication takes place.

Some researchers overlook the recipients' selection problem and assume that whenever an agent chooses to communicate, it broadcasts its message to all teammates (Goldman & Zilberstein, 2003; Roth et al., 2006; Reddy et al., 2012). Although some researchers adopted this strategy, it proved to be ineffective because recipients may not need the communicated information, or agents may communicate duplicated information, and hence waste communication resources.

The work by (Kinney & Tsatsoulis, 1998) assumed that information is classified into multiple classes. The authors proposed an approach to allow agents learn from each class of information, whether to communicate or not and to whom. This is achieved by having agents send feedback about received information to the sender. Moreover, (Roth, 2007) proposed an algorithm called Selective *ACE_PJB_COMM*, which works as follows. The algorithm takes as input two parameters k and n, where k is the maximum number of observations that can be communicated and n is the frequency with which agents can communicate. The algorithm starts with checking whether communication is allowed in the current time-step by comparing it with n, and if so, the algorithm uses a hill-climbing heuristic to choose at most k observations that are most beneficial if communicated. In the work by (Zhang, 2007), agents are classified to either provider of or needer for each information instance, hence each agent applies the utility function for each information to choose a communication strategy based on whether it is a provider or a needer. Furthermore, (Reddy et al., 2012) uses a game theory approach to address what agents should communicate by modeling communication as an extensive form game.

2.2.3 Communication Language and Protocol

Early research efforts in the field of communication in MAS were interested in formalizing and producing standards to facilitate agents' communication. This is important to allow heterogeneous agents that are developed by different designers and/or live in different environments to communicate. Most of the concepts used were inspired by research in human communication, of which the most fundamental is Speech Act Theory (Austin & Urmson, 1962). This theory explains how utterance is used to achieve goals and intentions, and it divides communication messages into different types such as inform, query, answer, request or command, promise or offer, acknowledge, and share. (Searle, 1976) introduced Indirect Speech Acts, which emphasizes the psychological interpretation of speech acts and classifies them as representative, commissive, directive, declarative, and expressive.

There are mainly two common standard languages for MAS that have been developed in the literature: Agent Communication Language (ACL), developed by the Foundation for Intelligent Physical Agents (FIPA-ACL) (for Intelligent Physical Agents, 2002), and Knowledge Query and Manipulation Language (KQML) (Finin et al., 1994). Both languages rely on Speech Act Theory (Searle, 1976). The main idea is that the different types of messages that agents exchange during communication are based on performatives proposed in Speech Act Theory (Chopra et al., 2013).

Different performatives have been proposed for each language, and for each message structure, a communication protocol is designed. Examples of FIPA-ACL performatives include but are not limited to inform, request, confirm, cancel, agree, propose, and refuse. A message in FIPA-ACL is structured to include parameters such as sender, receiver, content, reply-with, reply-by, in-reply-to, language, ontology, protocol, and conversation-id. Communication protocols define the flow of messages between two communicating agents for a specific purpose. More background, review, and critique of the literature of Agents' standard languages can be found in (Dignum, 2004; Dignum & Greaves, 2000; Singh, 2003)

What is more interesting than manually designing communication languages and protocols is to learn them. A large number of research efforts exist in the literature that propose and evaluate algorithms for learning communication language and protocols. For example, (Mackin & Tazaki, 2000) proposed an approach to evolve a communication protocol for negotiating agents using Genetic Programming (GP) in a multi-agent transaction system (e-commerce). The system consists of two types of agents: client agents and service provider agents, which negotiate and trade multiple different services. In their work, GP is used to evolve two programs for the negotiation process, message construction program (sender action), and messages handling program (receiver reaction). While a timer call is used to decide when the agents will execute the former program, receiving a message calls for executing the later program. If an agent decides to send a message, it can send it either to a specific agent or can use a pheromone model to leave a message object recognized by only agents in a specific surrounding domain. The fitness function of the GP computes the value of services received by the client agents. This work was extended in (Mackin & Tazaki, 2002) to evolve multiple negotiation protocols for agents with different objectives. Another example is the work by (Abdullah, 2005), where communication protocols are generated by learning performatives of ACL messages, performed in two steps. First, agents' conversations are transformed into markup agent communication language. Next, the language is used to develop communication protocols. Other research efforts have focused on learning a mutual language among agents. Examples include the work by (Giles & Jim, 2002; Froese, 2003; Khasteh et al., 2006b; Rawal et al., 2012; Gmytrasiewicz et al., 2002), where evolution is utilized to allow agents to learn a mutual language. A good source in this area is the work by (Bussink, 2004), which compares between communication protocols and language evolution in MAS.

2.2.4 Communication Interpretation and Ambiguity

This topic concerns the meaning of messages that agents exchange. Similar to humans, misunderstanding or misinterpretation of the intended meaning can occur between communicating agents, even if a mutual language exists. One reason that may cause misunderstanding is when agents are facing a new situation, such as recognizing a new object (Allen et al., 2005). Ambiguity can occur when a message carries multiple meanings. Existing of misinterpretation and/or ambiguity among agents may result in miscoordination and low overall performance (Goldman et al., 2004). Some works study the problem of messages misinterpretation. For example, the work by (Allen et al., 2005) relates to messages' meaning as the recipients' reaction to them. The authors compute the degree to which a message recipient understands the sender using messages that the receiver is likely to send. Specifically, the meaning of a message received is a distribution over the recipient's own messages. Moreover, in the work by (Goldman et al., 2004), the language spoken by the sender may not be completely understood by the recipient. Reinforcement learning is used to enable agents learn each other's languages. Each agent has a translation table, where rows are possible meanings and columns are received messages. Once an agent receives a message, it tries to interpret it by computing the probability distribution over possible meanings of its own language. Depending on the meaning, the agent takes an action and receives a reward, which is used to adjust the translation table. Furthermore, (Wang & Gasser, 2002) introduced a machine learning technique called Mutual Online Concept Learning (MOCL), where each agent can play a teacher, who teaches a concept, and a learner, who learns a concept. The learning framework consists of the following elements: concept (weight vector/function), instances (inputs to the concept), instance-producing mechanism (given a concept, produce an instance), instance interpreting mechanism (given instance, interpret using function to produce a value), and concept adaption mechanism (or online concept learning, given an instance from another agent, adapt the concept). The learning process takes place as follows. A teacher chooses a label in the range {-1,1}, produces a consistent instance, and sends it to the learner. Next, the learner predicts a label for the instance and sends it to the teacher, who in turn sends a feedback. Then, the learner updates his concept based on the feedback. In the work by (Kvasnicka & Pospichal, 1999), a listener agent decodes the received message and adapts its Artificial Neural Network (ANN) to decrease the difference between the speaker and listener meaning vector of the message. A Genetic Algorithm is utilized to evolve mutual pairwise communication.

In short, existing solutions for the misinterpretation and ambiguity problems in multi-agent systems are based on feedback from either a reward function or a teacher. When a reward function is used, usually interpretation of a message is viewed as the reaction of the agent to receiving a message, hence the receiver should interpret the message to the meaning that results in actions that maximize the reward function. However, when a teacher gives feedback, the specific meaning of the teacher should be predicted in order to achieve a successful interpretation.

2.3 Genetic Algorithm

Genetic Algorithm (Holland, 1975), is the most common algorithm in the Evolutionary Algorithm (EA) family. Unlike Genetic Programming (GP), where trees are used to represent solution candidates, GA's solution candidates are represented by a string of genes called chromosomes, which can take on binary, real, integer, or other forms of values. The representation of the chromosomes is problem-dependent, and can basically be in any format as long as it represents all properties of candidate solutions. Besides the genes' values, the position of the genes can be used to encode some features of the solution (as in, for example, (Wu et al., 2004)), hence permutations of the same chromosome represent different solutions. Moreover, depending on the problem, solutions can be of fixed or varying length.

There are mainly two different implementations of Genetic Algorithm (GA): generational and steady state (Noever & Baskaran, 1992). The main distinction between the two is the reproduction method. Generational GA replaces the whole or most of the population at each generation by performing a large number of crossover and mutation, which produce offsprings that replace individuals in the original population. However, steady-state GA replaces only a few individuals, usually two, by performing crossover and mutation and producing two offsprings (Syswerda, 1991). In the following sections, we explain each step in the evolution process using GA, and provide a short review of the literature for each step.

2.3.1 Principle of Evolution

Evolutionary Algorithm (EA) is a family of algorithms that borrow inspirations from the theory of evolution by (Darwin, 1859). Evolution is the slow and gradual modification of the ancestors' characteristics to generate descendants. A key mechanism in evolution is natural selection, by which better or fitted individuals survive, as opposed to mechanisms used in other learning algorithms, such as reinforcement (Blute, 1979). Evolutionary algorithms are defined as generic, population-based, meta-heuristic optimization algorithms (Bäck, 1996), which have been introduced to solve complex optimization problems. They are generic because, unlike other algorithms such as gradient descent that assumes differentiable function, EA makes no assumptions about the function landscape, and hence can be applied to any problem, (Potter, 1992). Moreover, EA are population-based since they evolve (or improve) a population of solution candidates, in contrast to algorithms that try to improve a single solution such as simulated annealing. EA includes multiple different algorithms that can be applied to problems of varying types. Examples include but are not limited to Genetic Algorithms (GA), Genetic Programming (GP), Evolutionary Programming (EP), and Evolution Strategy (ES). In the following section, we focus on GA as it represents our method to learn goal-driven communication strategies in MAS.

2.3.2 Process of Evolution

Initial population: The first step in running GA is to create an initial population, which consists of a set of individuals/chromosomes/candidate solutions. The number of individuals in a GA population can vary from a problem to another and can be set by the system designer. Research efforts exist in the literature that study the effect of varying the population size on GA's convergence and solution quality (Koljonen & Alander, 2006; Reeves, 1993; Roeva et al., 2013). A common practice in GA community is to find a good population size empirically (Eiben et al., 1999). Researchers in the past have argued that small population size can make GA converge faster but produce poor solutions (Koumousis & Katsaras, 2006; Pelikan et al., 2000; Piszcz & Soule, 2006), while large population size can prolong the evolution process but produce high quality solutions (Harik & Lobo, 1999; Lobo & Goldberg, 2004; Lobo & Lima, 2005; Koumousis & Katsaras, 2006; Rylander & Gotshall, 2002). However, this argument was negated by (Haupt, 2000) as it was found that running GA with small population size (as low as 16) can outperform large population size if the mutation rate is increased (to 5-20%). An offline adaptive approach, (Harik & Lobo, 1999), and online adaptive approach, (Arabas et al., 1994), were proposed in the literature to automate the process of selecting the best population size. In the former, the best size is found prior to running GA and remains fixed throughout the evolution process, while in the latter, the population size changes during the evolution process depending on some parameters such as the fitness value. More studies on how to choose the population size can be found at (Yu et al., 2006; Knaepkens et al., 2004; Smith & Smuda, 1995).

Besides the population size, generating the initial population is another issue that has been studied by the GA community. The common practice followed by the GA community is to generate the initial population randomly. However, researchers have argued that the quality of the initial population has a direct effect on the quality of solution found (Maaranen et al., 2007), and hence a number of metrics have been proposed to help measure the quality of the initial population. For example, (Diaz-Gomez & Hougen, 2007) proposed metrics to measure the diversity of the initial population at three levels: the gene level, the chromosome level, and the population level. (Goldberg et al., 1991) argued that there are building blocks (BB) that should be fed to GA in order to obtain solutions of high quality. Moreover, some efforts exist that apply different search algorithms to generate the initial population for GA, resulting in Memetic Algorithm (MA), (Devi et al., 2011). Examples include the work by (Kumar et al., 2013), where hill-climbing algorithm was used to generate the initial population, which shows to improve performance of typical Genetic Algorithm. (Kazimipour et al., 2014) conducted a survey of existing population initialization techniques, which were categorized with respect to randomness, compositionality, and generality.

Selection: After generating and evaluating the initial GA population, the evolution process starts by selecting parents for producing offsprings. This is a key step as it distinguishes GA from other learning algorithms by applying a biological theory called natural selection, which was introduced by (Darwin, 1859) as, “a principle by which each slight variation [of a trait], if useful, is preserved". In biology, this theory is observed as the likelihood of survival and reproducing of individuals or individuals' characteristics is more for those adaptive to the environment. In GA, this principle is translated, as the likelihood of survival and reproducing of solutions or sub-solutions is more for those fitted or better solve the problem. Therefore, solutions with high fitness are more likely to be selected as parents for producing offsprings because they are more likely to produce highly fit individuals. The fitness of a solution is evaluated using the fitness function, which is designed to measure the quality of the problem's solutions. The implementation of the fitness proportionate selection method is called roulette-wheel selection. A number of variations have been proposed in the literature for the selection method in GA and other evolutionary algorithms. Examples include truncation selection (Mühlenbein & Schlierkamp-Voosen, 1993), stochastic universal sampling (Baker, 1987), and tournament selection (Miller & Goldberg, 1995; Goldberg & Deb, 1991). Although in nature, mating and reproduction occur between two parents, researchers found that, for some problems, better solutions can be produced if multiple parents are involved (Elsayed et al., 2011; Patel & Raghuwanshi, 2010; Bonilla-Huerta et al., 2011).

Genetic operators: This represents the main means for reproduction of offsprings in genetic algorithms. In nature, a child inherits half of the genes from each parent, which are copied and combined to compose the child's DNA. Sometimes, small changes in the passed DNA may occur during the copy process, which result in the child having a bit different DNA strand than the parents. The occurrence of this natural modification is called mutation. Genetic operators in GA correspond to these natural processes, although multiple variations of each operator have been used in the literature (Syswerda, 1989; Caruana et al., 1989). The performance of GA is significantly affected by the choice of operators; hence great attention has to be paid to them during implementation (Yao, 1993). In this section, we explain the main genetic operators in GA, namely, crossover and mutation, and present some of the research efforts in the literature to improve them.

Crossover: is a process of combining sub-solutions (genes) from two solutions (chromosomes), where segments of genes are exchanged between parents to form offsprings. As have mentioned previously, it is analogous to how DNA strands from parents are combined and passed to children. In GA, however, two parameters have to be identified in order to implement crossover.

The first parameter is the *crossover point*, which specifies the point at which parents' chromosomes are divided and hence the number of segments exchanged. For sometime, the most common types were one-point and two-points crossover (Holland, 1975). However, some researchers reported that multi-point crossover can improve GA performance especially for ordered chromosomes, where the bit's position carries information about the solution (Eshelman et al., 1989; Syswerda, 1989; De Jong & Spears, 1992). This is due to the positional bias introduced by one- and two-points crossover (Eshelman et al., 1989). Crossover variants that can avoid this issue are the uniform crossover (Syswerda, 1989) and shuffle crossover (Caruana et al., 1989). With the uniform crossover, multiple crossover points may be used, which works as follows. For each gene in the offspring, a probability value (mixing rate) is used to determine whose gene, i.e., which parent, in the corresponding position will be copied. For example, if the mixing rate is 0.5, then each parent will contribute to 50% of the offspring's genes. With the shuffle crossover, however, only one crossover point is used. Positions of parents' genes are shuffled prior to the crossover and positions of offsprings' genes are unshuffled post the crossover. A comparative study between different forms of crossover can be found at (Magalhaes-Mendes, 2013; De Jong & Spears, 1992).

The second parameter is the *crossover rate*, which represents the probability with which a crossover occurs between parents at each generation. As crossover helps GA exploit known solutions (i.e., parents) to obtain close, but new, and hopefully better solutions (i.e., offsprings), crossover rate determines how much exploitation GA performs. One advantage of GA over other optimization algorithms is the balance between exploitation (crossover) and exploration (mutation, that is explained later), which can be controlled by tuning crossover and mutation rates (Eshelman et al., 1989). Hill climbing is an example of algorithm that is good at exploitation but makes little exploration (Vekaria & Clack, 1998). Research efforts to automate the process of optimizing the crossover operator are, for example, the work by (Lin et al., 2003; Vekaria & Clack, 1998; Schaffer & Morishima, 1987; Davis, 1989; White & Oppacher, 1994; Srinivas & Patnaik, 1994), where approaches for adaptive crossover operators are introduced.

Mutation: When offsprings are produced, they go through mutation, where some genes in the chromosomes are altered. As mentioned previously, genetic mutation is analogous to biological mutation where small changes may occur to DNA during the copy process. Multiple forms of mutation exist that depend on the solution representation of the problem. For example, flip bit mutation is a variation that can be applied to chromosomes with binary genes only, and in which a randomly selected bit is flipped (from 0 to 1, or 1 to 0). Other mutation forms are random and min-max mutation, which can be applied to chromosomes with integer or real genes only. The former exchanges randomly selected bits with random values in a specific range, and the latter exchanges a randomly selected bit with its minimum or maximum values.

Similar to crossover, *mutation rate* has to be tuned to optimize GA performance. Mutation goal is to preserve diversity during the evolution process as it helps GA explore new solutions. A common practice in GA community is to set mutation rate to a very small value in order to allow GA to converge, hopefully to a global maximum. A very high mutation rate can turn GA into random search (Vekaria & Clack, 1998). A good mutation rate is problem-specific and is usually found empirically. However, research efforts have tried to automate the process, such as (Libelli &

Alba, 2000; Zhang et al., 2010; Thierens, 2002; Blum et al., 2001; Korejo et al., 2009; Lin et al., 2003; Srinivas & Patnaik, 1994).

Termination: The process of evolution (selection, crossover, and mutation) continues until a termination criterion is met. A number of different termination criteria have been used in the literature. The most common one is setting a maximum number of generations (Koumousis & Katsaras, 2006; Lee & Yao, 2004; Lim et al., 2006; Ong & Keane, 2004; Ong et al., 2006; Tu & Lu, 2004; Yao et al., 1999; Zhou et al., 2007). Other common termination criteria are when a global maxima/minima is closely reached (Ong et al., 2006; Tsai et al., 2004; Zhong et al., 2004; Hansen & Kern, 2004), or when a predetermined number of generations do not improve the fitness (Leung & Wang, 2001). The work by (Safe et al., 2004) presents a critical analysis of the different existing termination criteria in genetic algorithms and highlights the main challenges associated with such decisions. The literature includes some research efforts to design heuristics that help GA decides when to stop. Examples include the work by (Bhandari et al., 2012), where a termination criterion was designed based on the variance of the best fitness values obtained during the evolution process. Moreover, (Hedar et al., 2007) proposed an Automatic Accelerated Termination (G3AT), where the amount of exploration that GA made is employed as a termination criterion.

2.3.3 Evolution of Communication

A wide range of research efforts exist in the literature that study the evolution of communication in software and robotic agents. Evolving a communication system has both advantages and disadvantages over using a predefined communication protocol (Bussink, 2004). Different researchers have different goals from evolving communication, such as demonstrating the capacity of Artificial Intelligence (AI), and understanding the origin of language and how human (or animal) communication has evolved. In a survey of existing work, communication systems may emerge from the interactions between evolutionary agents or robots, or evolved through artificial evolution.

In a pioneering contribution to the emergence of communication with the aim of showing AI capacity, MacLennan (Maclennan, 1990) emerged communication in a population of simple machines, detected by comparing simulations where communication was allowed with those in which it was suppressed. In that work, communication is stigmergic, as it refers to an agent putting information about its local environment in the global environment, which is accessible by all agents. Communication comes in different forms, including indirect/stigmergic communication, direct interaction, which involves observing teammates behaviors, and direct communication through messages passing (Trianni et al., 2004). An example of emergence of direct interaction is the work by Quinn (Quinn, 2001), where two simulated robots are assigned the task of coordinated motion, i.e., to move while staying close to each other and without colliding. The robots evolved a communication strategy such that the first robot aligns to face the other robot adopts the follower role, while the second to align becomes the leader. Direct communication has also been emerged in the literature (Marocco & Nolfi, 2005; Werner & Dyer, 1992; Werner & M, 1991; Baray, 1997). In the work by (Marocco & Nolfi, 2005), evolutionary robots were able to emerge a direct communication system based on five signals of different values that represent important features of the environment. In (Werner & Dyer, 1992), authors emerged communication in a population of female and male agents for the ability to mate. Female agents, which were deaf and immobile, evolved to emit signals when a male is in the same row or column to guide it to its position, whilst male agents, which were blind and cannot signal, evolved to turn when on the same row or column as a female. Other researchers, such as (Floreano et al., 2007; Nolfi, 2005; Wischmann et al., 2012), focused on investigating the evolutionary and non-evolutionary conditions required for the emergence of communication. Further, emergence of communication has also been applied to gain insight into the origin and evolution of communication. Examples include investigating the evolutionary origins of robots' communication system (Marocco & Nolfi, 2006a), and studying the development of a communication system in evolutionary robots that are initially non-communicating (De Greeff & Nolfi, 2010; Marocco & Nolfi, 2006b; Nolfi & Mirolli, 2009). The work by (Wagner et al., 2003) provides a review of computational models of communication emergence via learning and evolution, based on agents situatedness and whether communication is structured or not.

Due to the complexity of human language (Nolfi, 2013), research efforts that model its evolution usually focus on only parts of the language, such as evolution of signals, evolution of symbols (symbolic communication), and evolution of words and syntax. The work by (Cangelosi, 2001) explains the difference between words, signals, and symbols, and applies it to differentiate between models for the evolution of communication. It is explained in (Steels & Bleys, 2005) how the work had progressed to gradually cover elements of natural language. The work by (Marocco et al., 2003) made efforts to evolve signals and verbs in two separate experiments. A simple domain-specific language for the predator-prey pursuit problem was successfully evolved when predators communicate to a message board (Jim & Giles, 2000), and through message passing with one blind and another stationary predators (Rawal et al., 2012). More details on evolution of natural language can be found in (Cangelosi, 1999; Galantucci & Steels, 2008; Steels, 2003a,b, 2015).

Other aspects of communication have also been evolved in the literature. For example, evolution has been applied to learning a mutual language between agents (Giles & Jim, 2002; Froese, 2003; Khasteh et al., 2006a), evolving communication protocols (Gerard & Singh, 2013), evolving ontology such that each agent learns its own concepts (Afsharchi & Far, 2006), and evolving understanding between a group of agents, defined as the ability to accurately derive the internal state of other agents from their observable external state (Levin, 1995), or number of times that agents understand each other (Enee et al., 2004). The work by (Agah & Bekey, 1998) found out that agents tend to communicate more, and hence perform better, if their cognitive architectures are evolved.

2.4 Communication and Overall Performance

The root of research that studies the effects of communication on group performance can be traced back to the work about human groups (Leavitt, 1951) studying the influence of four different communication patterns (channels) on the overall performance and members behaviors. The communication patterns include circle, chain, Y shape, and wheel (or X shape), and they determine whom each team member can talk to. A simple data collation experiment was conducted, and it showed that the performance of the group and behavior of the members could be ordered as circle, chain, Y shape, and wheel, where the circle was the most erratic, had largest number of messages exchanged, and leaderless, but satisfying to its members, while the wheel was the least erratic, had fewer messages exchanged, and had a leader, but less satisfying to the members. This conclusion was quickly negated by the work of (Guetzkow & Simon, 1955), who stated that communication limitation only affects the way people organize themselves, and once this happens, they will have the same task performance. (Heise & Miller, 1951) conducted experiments with three different problems, similar to (Leavitt, 1951), but with the task declared complete when all members find the solution. They tested the performance of the group with five different communication networks, which determine directive communication channels. They found that the nature of the task plays an important role in determining the suitable communication network, and hence called for a way to classify groups' problems.

Multiple studies were published afterwards that proposed ways to classify tasks, such as (Shaw, 1973; Herold, 1978; Tushman, 1979; Steiner, 1972), of which one of the prominent ones is (McGrath, 1984), according to (Goodman, 1986; Stewart & Barrick, 2000). According to (McGrath, 1984), tasks can be classified according to the degree of which they are conceptual or behavioral. Consequently, tasks can be categorized to generating ideas and plans, choosing between alternatives, negotiating conflicts of interests, and executing work. Another related research effort is that by (Hirokawa, 1980), in which a study is conducted to compare communication patterns between effective and non-effective decision-making groups. The result states that a significant difference exists between the two groups, where effective decision-making groups spend more time discussing procedural matters and ensure that agreement is reached on substantial matters before starting a different discussion. (Mullen et al., 1991) studied the effect of centrality, in terms of degree, betweenness, and closeness, in communication on members' behavior of leadership, satisfaction, and participation. Results indicate that elements of centrality can be used to predict members behavior.

Recent research on communication and teams performance has focused on the effect of informal connections patterns or social networks structures on teams performance (Baldwin et al., 1997; Reagans & Zuckerman, 2001; Rosenthal, 1997; Wasserman & Faust, 1994; Sparrowe et al., 2001; Gloor et al., 2007). Conflicting results have been reported in the literature where some results indicate that social networks have impact on teams performance (Reagans & Zuckerman, 2001) and others indicate the opposite (Sparrowe et al., 2001). The work by (Balkundi & Harrison, 2006) carried out experiments to resolve the debate. Authors considered two different types of ties connecting individuals in a network according to the content exchanged: instrumental (formal relationship) and expressive (friendship) (Lincoln & Miller, 1979). The results indicate that better performance and more viable teams have been observed in teams with denser expressive and instrumental social networks (Balkundi & Harrison, 2006).

Successful project management has always been characterized by effective communication planning (Allen et al., 1980). A common practice in business organizations nowadays is to build a communication matrix/plan, which controls information flow within team's members and between team's members and other external parties such as stakeholders. The goal of communication plans is to deliver the right information (what), to the right people (who), at the right time (when) (Darnall & Preston, 2012). The importance of such practice comes from the fact that in the absence of communication plan, one of two situations could happen. Communication can be lower than required, and hence critical information can be easily missed or delayed, or more than required, where all available information is communicated to all parties, and hence recipients will be flooded with the large amount of information. This can prevent team members from concentration and make determining relevant information a difficult and time-consuming task. Developing a communication plan starts with identifying all parties who contribute to, are affected by, or can affect the project (Darnall & Preston, 2012). This can include individuals inside the organization such as team members, leaders, and colleagues, or others outside the organization such as stakeholders, customers, sponsors, vendors, and professionals (Kirchenbauer, nd). The second step is to determine information that needs to be provided to each party (Darnall & Preston, 2012). Examples include strategy changes and updates, timelines and deadlines, and technical reviews (Kirchenbauer, nd; Darnall & Preston, 2012). The third step is planning when such information should be communicated to recipients; such as daily, weekly, monthly, or quarterly (Kirchenbauer, nd). Moreover, communication planning can as well include how communication is carried out, such as through reports, physical meeting, phone call, e-mail, social media, an event, or online meeting (Kirchenbauer, nd). Table 2.1 shows an example of a communication plan.

Table 2.1: An example of communication plan/matrix.

Information	Recipients	Frequency	Means
Project progress	Project's manager	Weekly	Meeting
Project risks	Business manager	As needed	Conference call
Project update	Project's team	As needed	E-mail
Technical issues	IT manager	As needed	E-mail
Funding requests	Sponsors	Monthly	Phone call
Schedule changes	Project's team and manager	As needed	E-mail
Project Status	Steering committee	By 2 p.m. every Mon.	Meeting

In Multi-agent Systems, the related work that links communication and performance can be broadly classified into two main categories: (1) those that address a specific performance problem using communication, and (2) those that address communication needs according to a single predetermined performance goal.

2.4.1 Communication as A Solution

(Kinney & Tsatsoulis, 1998) emphasized in their work that variety of performance problems in distributed artificial intelligence and multi-agent systems, such as idle time, work duplication, and coherency, can be reduced to addressing communication. Further, they argued that research addressing other MAS's problems, such as agents' topology, coordination, and knowledge representation, are in essence addressing communication.

Examples of existing work that belong to the this category include using communication to address uncertainty (Xuan & Lesser, 2002), agents' idle time (Clair & Matarić, 2011), load balancing (Yahaya et al., 2011; Bigham & Du, 2003), and coordination (Szer & Charpillet, 2004). Other researchers used communication to improve policies computation and representation (Nair et al., 2004), decrease online planning complexity (Wu et al., 2011), and learning social rules and compensating for limited sensing (Mataric, 1995). Another research effort is that of (Dutta et al., 2005), which proposed a post-task-completion communication protocol to improve distributed learning of the states of other agents.

The limitation of this direction of research is that the proposed communication approaches are exclusively designed to improve or optimize the performance of the system with respect to the particular performance goal in consideration. Therefore, if a system suffers from multiple performance problems or needs to improve its performance with respect to a different goal/metric in different times or situations, a different approach needs to be taken for each case.

2.4.2 Communication as A Problem

Our work is more related to the second category, where research efforts are directed towards addressing the tradeoff between performance and communication cost. Despite the variations in modeling and designing the system, a rule of thumb is that information is communicated only if doing so is beneficial to the system's performance (Becker et al., 2009), based on Information Value Theory (Howard, 1966). Many approaches have been proposed in the literature to determine when communication is beneficial. For example, research efforts, such as (Chakraborty & Sen, 2007; Melo & Spaan, 2011; Roth, 2007), consider communication beneficial if the difference between the expected performance improvement with communication and the expected improvement without communication is larger than the communication cost. (Ghavamzadeh & Mahadevan, 2004) compares the expected value of communication plus communication cost with the expected value of not communicating. Value Of Communication (VOC), computed as the difference between the expected value when communicating and the expected value for remaining silent, is applied in (Becker et al., 2009; Carlin & Zilberstein, 2009; Tian et al., 2013), where communication is allowed only if VOC>0. Other works, such as (Williamson et al., 2008; Dutta et al., 2007), propose approaches based on Kullback-Leibler (KL) divergence and information redundancy, respectively. Zhang (Zhang, 2006) considers two factors to value information: timeliness (using the most recent received but potentially outdated value), and relevance (waiting for the new value). The work by (Unhelkar & Shah, 2016) proposes ConTaCT, an online communication decision algorithm. At each time step, each agent computes three rewards; the team's reward using previously chosen policy, the team's reward using locally modified policy and no communication of the modification, and the team's reward using globally modified policy and communication of an observation. An agent decides to communicate only if the latter reward is larger than the former two rewards. In this work, a fitness function is designed to balance between the system performance with respect to a given goal and communication cost, according to flexible weights.

2.5 Communication with a Goal

The notion of goal-driven (or goal-oriented) communication has been proposed and studied in the literature. Researchers, however, have used the term goal to refer to varying concepts. For example, (Goldreich et al., 2012) proposed a mathematical theory of goal-oriented communication, where a formal definition of the communicating parties and goals is given. In their study, communication is considered as a way to accomplish goals of communication parties in a user-server setting, such as computer-printer, with no mutual protocol/language. Goal is defined as “the way we (the users) wish to affect the environment and/or to the information we wish to obtain from it" (Goldreich et al., 2012). In particular, the authors proved that if a notion of (sensing) is available, which allows the communicating parties to check whether a progress has been made towards the goal as an affect of communication, then no priori mutual language is required; rather there exists universal user strategies that are able to accomplish the goal. As the approach can suffer exponential overhead, work by (Juba & Vempala, 2011) proposed an approach that can reduce the complexity to polynomial.

In (Juba, 2011), the author distinguished between finite and infinite goals, where the former represents an agent's desire to reach a state of being, and the latter represents the agent's desire to maintain a state of being. Moreover, (Van Oijen et al., 2011) designed a goal-based dialogue system using BDI agents, where an argument was developed, which states that ontology can structure information exchange between agents as well as the agents' knowledge. In their work, a communication goal is defined as a single ontology-driven dialogue between two agents, of which examples are persuasion/informing, information seeking/querying, and deliberation/ordering.

Furthermore, (Cheong & Winikoff, 2005) proposed a goal-oriented approach to interaction protocol between agents, which is unlike other protocols where no messages sequence is defined, rather agents use interaction goals, available actions, and constraints to define them. Goals are defined to be of the interaction not agents; hence an interaction is declared successful if its goals are achieved. In addition, every goal can be broken down into smaller goals, which can be connected using temporal dependencies that add order constraints on executing the interaction. The work by (Braubach & Pokahr, 2007) proposed a goal-oriented communication protocol, where authors distinguished between two levels of interaction goals, namely, macro and micro. In the macro level, the overall goals of the system need to be identified, whilst in the micro level, every agent's goal is identified.

2.6 Limitations of Current Approaches for Addressing Communication

Communication decisions in cooperative MAS can be either centralized, where a coordinator agent that has a full observation of the global state computes a central strategy, or decentralized, where each agent with local observation computes its own strategy. In the latter case, however, cooperative agents need access to states and actions of their teammates to be able to estimate the overall communication benefits, and hence make good communication decisions. Therefore, researchers have designed different approaches to allow agents obtain such information. For example, (Kinney & Tsatsoulis, 1998) allowed agents to send feedback about the usefulness of the information they received to senders, (Roth et al., 2005) allowed agents to take actions based on shared information, and hence know the actions taken by the teammates, and (Zhang, 2006) extended agents' observability to enable agents to track team members' mental states, and hence infer what teammates know and when. Based on the domain characteristics, some research efforts, such as (Roth et al., 2005, 2006; Tian et al., 2013; Williamson et al., 2008), proposed approaches based on modeling the team's decision problem using variations of Markov Decision Process (MDP), such as Dec-MDP and Dec-POMDP, to enable agents to estimate the impact of communication, and hence compute their communication policies. The computation, however, is usually based on myopic assumptions, where each agent evaluates the benefit of communication in isolation for 1-step horizon assuming others never communicate. The work in (Becker et al., 2009; Carlin & Zilberstein, 2009) proposed approaches that relax these assumptions, yielding better performance. As mentioned previously, the work presented in this dissertation addresses this issue by adapting a centralized learning offline and distributed execution at runtime.

Further, most of the works, presented in this chapter, address only part of the communication decisions in MAS. For example, (Becker et al., 2009; Carlin & Zilberstein, 2009; Chakraborty & Sen, 2007; Dutta et al., 2007; Ghavamzadeh & Mahadevan, 2004; Iba et al., 1997; Melo & Spaan, 2011; Roth et al., 2005; Williamson et al., 2008) address only when agents communicate, (Roth et al., 2006) addresses what agents communicate, and (Goldman & Zilberstein, 2003; Tian et al., 2013; Zhang, 2006), address what and when agents communicate. Similar to our work, (Kinney & Tsatsoulis, 1998) addresses what, when, and to whom agents communicate. However, authors provided agents with only one timing strategy, which is frequency of communication, equivalent to one of our timing strategy, EveryTimeInterval, which will be explained in a later section.

Different metrics have been used in the literature to value communication decisions of agents. Examples include performance-based metrics such as minimum communication cost (Kinney & Tsatsoulis, 1998; Zhang, 2006), task progress (Williamson et al., 2008), minimum time (Goldman & Zilberstein, 2003; Iba et al., 1997), and avoiding coordination errors (Roth et al., 2005). Also, information-based metrics have been used such as timeliness and relevance (Zhang, 2006), information redundancy (Dutta et al., 2007), and KL divergence (Williamson et al., 2008). Some works, such as (Bouslimi et al., 2014; Gutiérez et al., 2009), proposed metrics to evaluate the quality and balance of communication in multi-agent systems. The advantage of the proposed approach in this dissertation over existing ones is that it is designed with no assumptions about the desired system's performance. Therefore, system designers have the ability to develop their own metric, with which they would like to improve the system's performance, with the desired trade-off with communication. This contrasts with an implicit assumption made by researchers in this area, which suggests that the system's performance has to always be improved with respect to a single predetermined goal.

CHAPTER 3

Goal-Driven Communication

3.1 System Design

IN THIS work, the learning problem of goal-driven communication is to find a communication strategy that (1) improves the global performance of the system with respect to flexible performance goals, and (2) achieves a good user-defined trade-off between the performance and communication cost. Unlike most of the research efforts that address communication, our work does not aim to enable agents to reason about their communication decisions by valuing each communicating decision in isolation. Rather, we value a communication strategy and aim to develop a mechanism for learning a central strategy at plan time that, when followed by agents in a decentralized manner, achieves a global behavior in favor of a specific performance goal. We achieve this by utilizing Genetic Algorithm, and designing a fitness function that values a given communication strategy according to the two aforementioned factors.

Definition: a Communication Strategy (*CS*) is a set of m sub-strategies that determines what information is to be communicated, when, and to whom, such that $0 \leqslant m \leqslant (k * c)$, where k is the number of information instances, and c is the number of agent types. Each sub-strategy is a communication rule for one information instance (e.g., IS_i) that determines whether IS_i is communicated, and if so to whom (recipients) and when (timing), it is communicated.

Other important notions that should be distinguished are information instances and value updates of information instances. For example, target's position is an information instance, but an observed target's position ,e.g., (1,1), is a value. The information instances that agents communicate during task execution can be classified into two categories:

1. **Single-value**: Information instances that, at any time-step, an agent is allowed to have only one value. Examples include agents' goals since an agent can have only one goal at any point of time.

2. **Multi-value**: Information instances that an agent can possess multiple values, such as objects' locations since an agent may have a list of all locations where it observed an object.

3.1.1 Recipients Strategy

This strategy determines the recipients of value updates of an information instance IS_i. When multiple types of agents exist, this strategy defines the recipients group (e,g., G), as well as the number of recipients from the group based on the proximity of the recipient. The following strategies are considered (illustrated in Figure 3.1):

Peer-to-Peer (P2P): Allows communicating value updates of the information instance to only the closest agent, from a group G, to the sender.

Subset $(\boldsymbol{x = [2-(n-1)]})$: Allows sending value updates of the information instance to the closest x agents, from group G, to the sender, where n is the total number of agents.

Broadcast (Bcast): Allows communicating value updates of the information instance to all agents from group G.

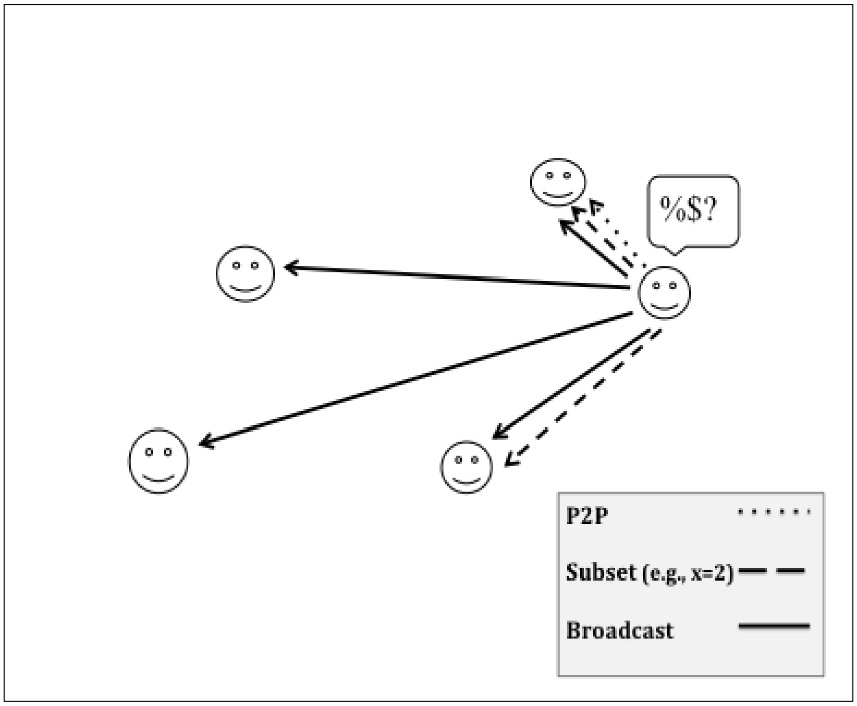


Figure 3.1: Recipients Strategies.

3.1.2 Timing Strategy

The timing strategy corresponds to the time-steps, at which value updates of an information instance are communicated. Three timing strategies are considered, and are illustrated in Figure 3.2:

Every update (EU): Agents are allowed to communicate at all time-steps, and hence value updates are communicated at the same time-step that they are obtained. In case of single-value information instances, a new value update is communicated once obtained; whilst for multi-value information instances, agents communicate whenever a new value is added to the values list of the information instance. This strategy is ideal for information instances that need immediate response, or reaction from others, or when delay is not tolerated.

Every time interval (TI=[2-i]): Agents are allowed to communicate only at specific time-steps, separated by an evolved time interval. For instance, if TI=3, then agents are allowed to communicate value updates of an information instance every 3 time-steps. If the information instance is single-value, an agent will check every 3 time-steps and communicate only if the value is updated since the last communications time. Therefore, if multiple updates occur before the next communication time, only the last update is communicated. However, if the information instance is multi-value, then all newly added values for the information instance are communicated. One advantage of this timing strategy is that it is more communication efficient, as it combines multiple values into one message.

In a state: Agents are allowed to communicate only value updates that are obtained when the agent is in a specific state *s*, hence communicating only specific values of the information instance. An advantage of this strategy is that it is ideal when only specific values of an information instance can contribute to improving the performance. For instance, in work by (Balch & Arkin, 1994), robots performing forage task are only interested to know if another robot is in the *acquire* state, because they found useful work, and hence if others follow them, they will potentially find useful work too.

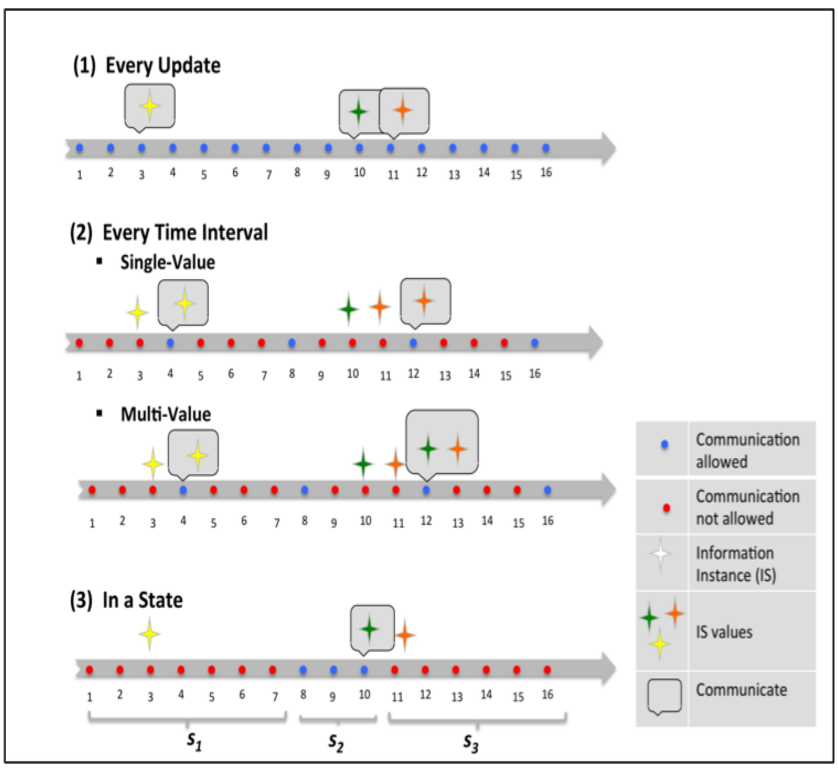


Figure 3.2: Illustrations of timing strategies.

3.2 Problem Formulation

The problem can be formulated as: Given a set *IS* of *k* information instances and a performance goal *P*, find a communication strategy *CS* that achieve a good trade-off between the communication cost and the system's performance with respect to *P*.

We shall evaluate the complexity of the search space. For an information instance IS_i, it is assumed that there are *t* alternatives for when it can be communicated, and *h* alternatives for who it can be communicated to. Therefore, the total number of distinct sub-strategies *S* is:

$$|S(ISi)| = t * h + 1$$

The number of alternatives for the when component multiplied by the number of alternatives for the who component plus the option of not communicating the information instance. Hence, for a set *IS* of *k* information instances, and *c* types of agents, the total number of possible communication strategies (*CS*) can be computed as:

$$|CS(IS)| = (t * h + 1)^{k*c}$$

The number of possible communication strategies increases exponentially with the number of information instances *k* and number of agent types *c*. Besides, due to the non-determinism in multi-agent systems, it is usually not clear upfront which communication strategy will be effective with respect to the task and performance goal (Preist & Pearson, 1998; Balch & Arkin, 1994; Wei et al., 2014), and it is difficult and time-consuming to manually try all possible communication strategies. Yet, combinations of different sub-strategies for each information instance may result in unexpected performance. This calls for an automated approach for determining an effective communication strategy with respect to the performance goal, which is proposed in this research. We only consider explicit communication by sending and receiving messages. We assume that a communication language already exists, and that communication is always reliable.

3.3 Algorithmic Approach

We use steady state genetic algorithm to evolve goal-driven communication strategies. Unlike generational GA, in which a large portion of the population go through crossover and mutation and replaced by new offsprings, in steady state GA only two parents are selected from the population and crossed over generating two offsprings, which are then possibly mutated and placed in the population to form the new generation. A flowchart of the general steps to evolve goal-driven communication is depicted in Figure 3.3.

3.3.1 Solution Representation

A communication strategy, represented as a GA individual, is defined to be a vector of m cells. A cell, representing a sub-strategy, is a 4-tuple vector $(\boldsymbol{IS_i}, \boldsymbol{G}, \boldsymbol{R}, \boldsymbol{T})$, which defines when, and to whom value updates of an information instance are communicated, where:

$\boldsymbol{IS_i}$ is an identifier for the Information instance

$\boldsymbol{G}$ defines the recipients agent type

$\boldsymbol{R}$ is recipients strategy

$\boldsymbol{T}$ is the timing strategy

$\boldsymbol{m}$ is the number of communicated information instances, and $m = [0, k * c]$.

The first two elements, $\boldsymbol{IS_i}$ and $\boldsymbol{G}$, combined identify a cell, which indicates that value updates of an information instance $\boldsymbol{IS_i}$ are communicated to $\boldsymbol{R}$ agents from $\boldsymbol{G}$, only at time-steps that conform to $\boldsymbol{T}$. The length of a communication strategy may range from zero, which means that communication is not allowed at all, up to $k * c$ cells, which means that the strategy allows communicating value updates of all k information instances to all c agent types. This flexibility allows learning what information instances are communicated that would improve the stated performance goal.

Table 3.1: Possible values for each token in a cell/sub-strategy.

Token	Values	Notes
IS_i	[1, k]	k: number of information instances in the domain.
RT	[1, c]	c: number of agent types in the domain.
R	P2P (1) Subset [2, (n-1)] Broadcast (n)	n: total number of agents
T	EveryUpdate (1) EveryTimeInterval [2, i] InState [(i+1), (i+s)]	i: maximum time interval s: number of possible agents states

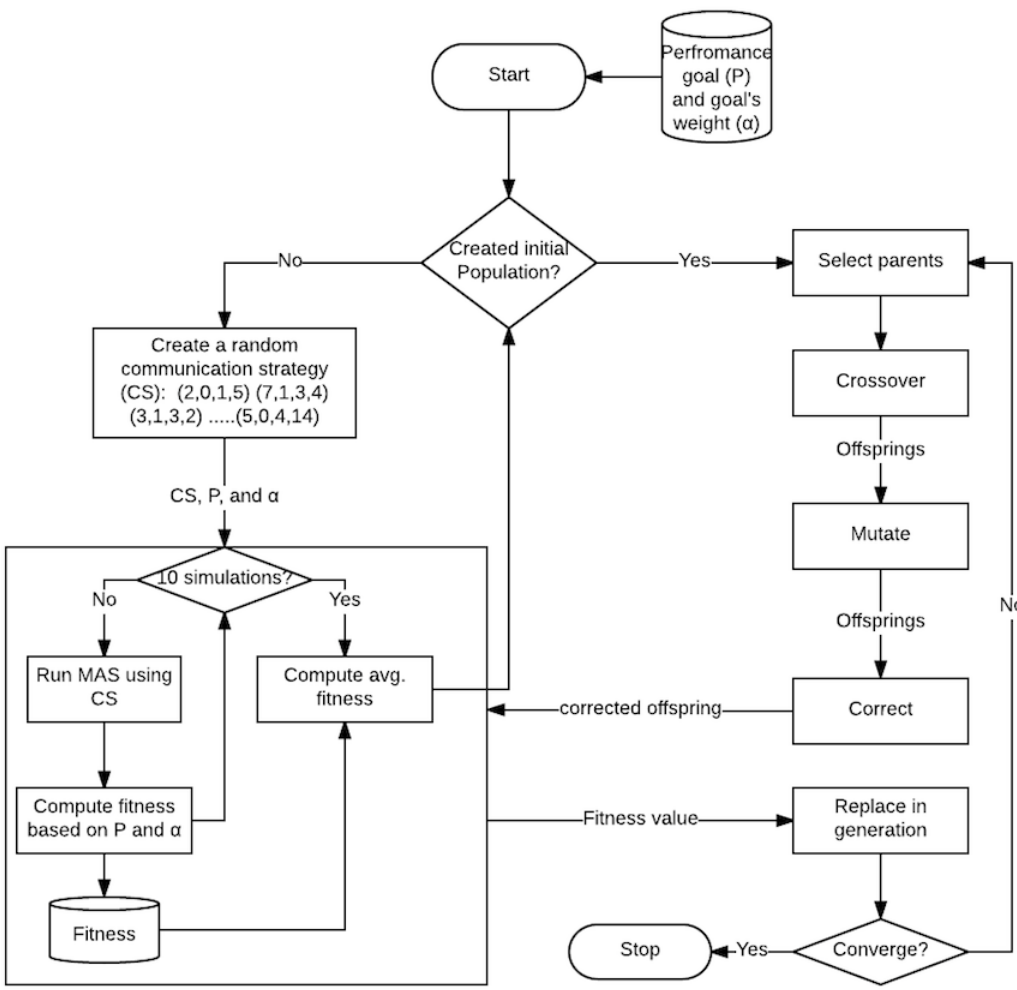


Figure 3.3: Evolution of goal-driven communication strategy.

3.3.2 Fitness Function

Despite the desired performance goal, there are two factors that contribute to the fitness of a communication strategy. The first factor is performance of the system using the strategy with respect to the goal, and the second is the strategy's communication cost. Since higher values for the performance goals that are considered in this work imply greater utilization of resources, we consider both factors to be costs; i.e., goal cost and communication cost, hence for both lower values implies better fitness. Therefore, the fitness function is designed as a weighted sum of the two costs, where the weight assigned to each cost reflects the designer's priorities. The GA goal is to minimize the total cost. Higher weight assigned to a cost implies greater contribution to the final fitness, and hence higher priority of being minimized. Further, since the two costs have different units and may not be directly summed, each cost is divided by the maximum value, which is obtained empirically from a large number of simulation runs, hence bringing the scale of each cost to [0,1]. We, further, multiply each cost by 100, as shown in Eq. 1.

$$fitness = \alpha\left(\frac{goal\_cost}{max\_goal\_cost} * 100\right) + \beta\left(\frac{comm\_cost}{max\_comm\_cost} * 100\right) + p \qquad (1)$$

Where,

α: is weight for goal cost.

β: is weight for communication cost, and $\beta = 1 - \alpha$.

p: is domain-based penalty term, for example $p = 1,000 * objects\_left$ in a forage task.

The penalty term is necessary to avoid evolving strategies with good fitness that do not allow the agents to complete the task. In order to minimize variance across simulation runs, the fitness is computed as the average fitness of n runs as follows.

$$avg\_fitness = \frac{1}{n}\sum_{i=1}^{n} fitness \qquad (2)$$

3.3.3 Genetic Operators

Additional operators as well as modification to some existing operators are required for the GA to produce valid individuals/communication strategies. This section illuminates the main genetic operators used in this approach and how our adopted modifications contributed to better performance.

Crossover: Since each solution candidate consists of multiple cells corresponding to sub-strategies for different information instances, a special crossover operator that swaps a subset of cells between two parents is needed. Allowing crossover point to only take place between cells and never divide a cell can enforce this (see Figure 3.4). This special crossover operator has been used previously in another work (Wu et al., 2004), where a cells-like GA individual representation has been used. This has the advantage of producing valid offsprings, as well as preserving GAs property of maintaining and using successful cells, a.k.a., sub-solutions or sub-strategies, found in previous generations as a building block to discover new valid solutions. In this work, we use one-point crossover, where the crossover point is randomly chosen for each parent. Figure 3.4 shows an example of two parents going through crossover.

Mutation: When two offsprings are produced after crossover, they go through mutation. While crossover helps GA exploit the good solutions found so far, mutation makes sure that GA explores new solutions. In this work, mutation rate is the probability that one token of a cell will be changed.

Selection Many approaches have been proposed in literature for GA selection method (Goldberg & Deb, 1991). In this paper, roulette wheel selection is adopted as our selection method. Roulette wheel selection is a fitness proportionate selection method, in which fitted individuals are more likely to be chosen to produce new offsprings, since they have more potential to produce highly fitted individuals.

Replacement: When new offsprings are produced, a good replacement strategy must be used to choose which individuals in the population are eliminated to make spots for the new individuals. A trivial strategy could be eliminating the least fit individuals in the population. However, this

First Parent with crossover point 2: (7,1,3,7) (8,0,1,23) | (4,0,3,18)
Second Parent with crossover point 1: (6,1,4,5) | (2,0,1,2) (1,0,4,14) (7,1,4,4)

First Off-Spring (redundant cells detected): **(7,1,3,7)** (8,0,1,23) (2,0,1,2) (1,0,4,14) **(7,1,4,4)**
=> Cancel a redundant cell at random: ~~(7,1,3,7)~~ (8,0,1,23) (2,0,1,2) (1,0,4,14) **(7,1,4,4)**
Second Off-Spring: (6,1,4,5) (4,0,3,18)

First Off-Spring: (8,0,1,23) (2,0,1,2) (1,0,4,14) (7,1,4,4)
Second Off-Spring: (6,1,4,5) (4,0,3,18)

Figure 3.4: Crossover and correction operations.

strategy will quickly expel diversity out of the population (Luke, 2013; Gupta & Ghafir, 2012). A replacement strategy for steady state genetic algorithm has been proposed in (Lozano et al., 2005) that takes into account two factors, namely, diversity contribution to the population and fitness function. In this work, we follow (Thierens, 1997) in assuming that parents are the most similar individuals to the new offsprings; and therefore, the four individuals, two parents and two offsprings, compete for insertion in the population. The best two out of the four individuals are inserted in the population in order to form the new generation.

Correction: When subsets of cells are exchanged during crossover to form new offsprings, it is likely that the new offspring could contain duplicated cells. The duplication does not necessarily mean that the two cells are exactly the same, but two cells define communication strategies for the same information instance and to the same recipient type, i.e., the first two elements of the cells are the same, such as; (2,0,5,3) and (2,0,3,1). If this occurs after crossover and mutation, one of the duplicated cells will be chosen randomly and deleted.

CHAPTER 4

Research Methodology

WE ADOPT an experimental methodology to evaluate the effectiveness of the proposed approach (Althnian & Agah, 2015). For this purpose, we conduct experiments on two different, well-known multi-agent testbeds, namely, The Wumpus World and Collective Construction . Following, we briefly introduce the two case studies. A detailed description is provided in the next two chapters.

4.1 Case Studies

4.1.1 The Wumpus World

The Wumpus World is originally a computer game called Hunt the Wumpus, developed by Gregory Yob (Yob, 1975). The game became a popular testbed for intelligent systems and is discussed in the leading AI book (Russell & Norvig, 2003), after a suggestion by Michael Genesereth.

In this work, we develop a multi-agent version of the Wumpus World Problem, similar to (Zhang, 2006), where a team of carriers and fighters cooperate to collect gold and kill wumpuses, present in the environment. Further, we enhance the simulation environment to include several rooms that contain the gold and wumpuses. This new feature provides a variety of information instances, which adds a key property that makes the domain effective for evaluating communication strategies.

This problem domain is closely related to a class of real-world application domains such as First Responders on a mission to rescue victims of natural or wars disasters. It is a common practice in AI and Robotics to use games to test the efficacy of newly proposed approaches, although the goal of research has always been solving real-life and complex problems. The rationale is that testing and application of algorithms to real-life problems is expensive and difficult, while games are easy to explain, understand and implement, and more importantly have direct correspondence to real-life problems as they are designed to encode all the challenging aspects of real-life problems. For instance, in case of the Wumpus World, rooms correspond to different buildings and locations, while the drop-off room corresponds to the ambulance or emergency rooms. In addition, gold represents victims to be rescued, while wumpuses are obstacles or hazards. Finally, carriers and fighters are the rescuing individuals and special-task individuals, such as emergency personnel, respectively. More details about the Wumpus World domain is provided in Chapter 5.

4.1.2 Collective Construction

The origin of the Collective Construction domain is inspired by the blind bulldozing behavior found in certain species of social insects, such as wasps and ants (Werfel & Nagpal, 2006; Parker & Zhang, 2002), where a colony use mud and wax to build a 3-dimensional nest. We adopt a simple version of this domain, where a group of different software agents cooperate to build a user-defined 2-dimensional structure using square blocks. Automated construction systems have been of interest to reseachers recently as it is useful in environments where traditional construction methods that involve human cannot be applied (Meng & Jin, 2012; Werfel & Nagpal, 2006; Wawerla et al., 2002; Schuil et al., 2006; Petersen et al., 2011). The reasons are that human involvement could be inconvenient or prohibitively expensive, such as in extraterrestrial or underwater environments, or dangerous such as in disaster areas (Werfel, 2012). Many researchers have described this problem domain as crucial to the success of robotic missions to the planet Mars (Parker et al., 2003; Parker & Zhang, 2006).

In this work, three types of agents are developed, namely, bulldozers, collectors, and builders, to perform three main tasks required for the construction of a structure, including (1) preparing the construction site; (2) searching for, collection, and transportation of the building materials to the construction site; and (3) building the structure by sorting the building materials into the proper form, respectively (Parker et al., 2003). More details about the Collective Construction domain is provided in Chapter 6.

4.2 Evaluation Criteria

The following three criteria are used to evaluate and compare the learned goal-driven communication strategies:

1. Performance of the system with respect to the stated goal using the learned communication strategy.

2. Communication cost of the learned communication strategy.

3. Fitness of the learned communication strategy with respect to different performance goals.

4.3 Experimental Design

4.3.1 Fitness Parameters

The first set of experiments focuses on finding a communication strategy that, for a given goal, achieves a customized trade-off between two desirable but incompatible costs: a goal cost and communication cost. Evidently, a pressing question arises in this situation: What is the relative importance of each cost? This concept is expressed in weights associated with each cost, namely α for goal cost and β for communication cost, in the fitness function. We consider three cases: (1) goal cost is more important than the communication cost, (2) communication cost is more

Table 4.1: Fitness parameters settings

Costs	**Weights**	α	β	**ac**	**cc**
ac=cc	$\alpha<\beta$	0.25	0.75	1	1
	$\alpha=\beta$	0.50	0.50	1	1
	$\alpha>\beta$	0.75	0.25	1	1
ac<cc	$\alpha<\beta$	0.25	0.75	1	2
	$\alpha=\beta$	0.50	0.50	1	2
	$\alpha>\beta$	0.75	0.25	1	2
ac>cc	$\alpha<\beta$	0.25	0.75	2	1
	$\alpha=\beta$	0.50	0.50	2	1
	$\alpha>\beta$	0.75	0.25	2	1

important than goal cost, and (3) both costs are of the same importance.

If the goal cost is more important than communication cost, we assign a high value to α (>0.5). This applies to situations when the goal performance is critical, which demands minimization of the associated cost to the lowest possible value. This can, in turn, cost more communication that is enabled by assigning a lower value to β (1- α). This conveys the fact that communication is not free; hence the system's designer is willing to allow high communication, but still desires to limit it to only what could lower the goal cost (hence, remove unrelated communication). For the second case, communication has higher priority than the goal performance. Therefore, α is assigned a low value (<0.5), whereas β is assigned a high value (1- α). The third case lays in the area between the previous two cases, where the goal performance and communication cost have equal priority, hence both α and β are assigned the same value (0.5). Besides the weights, another key factor that we studied is the action cost (ac) and communication cost (cc). Actions include moving, picking up and dropping off gold, and killing a wumpus. Similar to weights, we consider three cases: (1) action cost is greater than communication cost, (2) communication cost is greater than action cost, and (3) both costs are equal. Therefore, fitness parameters include weights {α, β} and costs {ac, cc}. Combination of the aforementioned variations of costs and weights result in nine settings (Table 4.1), which we carried out for each performance goal.

4.3.2 Fitness Goals

Since the purpose of this work is evolving goal-driven communication strategy, it seems natural to run experiments with different goals. The second set of experiments focuses on how changing the performance goal, for a given parameter's setting, can affect the evolved strategy and hence the performance. The desired performance goal is expressed in the GA fitness function. In this work, four goals are considered, specifically, time, travel distance, effort duplication, and energy.

Time measures the number of time-steps needed to complete the task. In the Wumpus World domain, the task is considered completed when the last piece of gold in the environment is collected, while in the Collective Construction domain, the task is finished when the structure is completely built. This metric can be used if the user needs the task to be completed as fast as possible.

Travel distance measures the total number of moves, performed by all agents, until the task is complete. A move is defined as changing position. This metric can be used if the user needs the task to be completed with minimum travel distance.

Effort duplication occurs when two agents target the same goal at the same time, or when one performs a previously completed work.

For the Wumpus World domain, this metric measures the duplication of efforts for carriers and fighters. For carriers, it measures the number of times two or more carriers explore a room at the same time, and number of times a carrier fails to pick up gold because someone else has already collected it. For fighters, it measures the number of times two or more fighters target the same wumpus.

For the Collective Construction domain, this metric measures the duplication of efforts for builders, bulldozers, and collectors. For builders, it measures the number of times a builder fails to attach a block to a structure site because it is already occupied, and when a builder fails to pick up a block from a supply zone because it is already collected. For bulldozers, it measures the number of times a bulldozer fails to remove debris because it has already been cleared. For collectors, it measures the number of times a collector fails to pick up a block because it is already collected, and the number of times it fails to place a block in a supply zone because it is full.

Energy measures the total amount of energy consumed by all agents to finish the task. Agents consume energy when they perform actions, such as moving, picking up and dropping off an object, and when they communicate. This performance metric can be used if the user needs the task to be completed with minimum energy cost. Unlike the previous goals, energy is a single-objective goal as communication cost is already included in energy.

4.3.3 Simulation Environment

There are many variations that could be applied to the simulation environment in order to test the proposed approach in different scenarios. We study the impact of variations of two features, namely Task Complexity (TC) and Agent Population (AP), which constitute a sufficient representation of the simulation environment.

The Wumpus World: Task complexity is defined as the number of wumpuses and amount of gold present in the environment, whilst agent population is the total number of carriers and fighters. Variations in these two features on this domain can be classified into three categories, as shown in Table 4.2. These include variations where (1) Task complexity is greater than agent population, (2) Task complexity is lower than agent population, and (3) Task complexity is equal to agent population.

Collective Construction: Task complexity is defined as the size of the structure, number of blocks, and amount of debris present in the environment, whilst agent population is the total number of collectors, builders, and bulldozers. Variations in these two features on this domain are shown in Table 4.3.

Table 4.2: Variation in the Wumpus World simulation environment.

	Agent Population (AG)			**Task Complexity (TC)**		
	Carrier (C)	Fighter (F)	Total	Gold (G)	Wumpuses (W)	Total
Basic Case	5	3	8	10	6	16
TC>AG	5	3	8	20	12	32
	5	3	8	30	18	48
TC<AG	15	9	24	10	6	16
	20	12	32	10	6	16
TC=AG	10	6	16	10	6	16
	5	3	8	5	3	8

Table 4.3: Variation in the collective construction simulation environment.

	Agent Population (AG)			**Task Complexity (TC)**		
	Builder (B)	Bulldozer (Bz)	Collector (C)	Blocks (Bk)	Obstacles (O)	Structure Size (SS)
Basic Case	3	5	5	200	100	10
TC+	3	5	5	300	150	15
	3	5	5	400	200	20
	3	5	5	500	250	25
AG+	6	10	10	200	100	10
	9	15	15	200	100	10
	12	20	20	200	100	10

4.4 Experimental Setup

In order to compute the fitness of one communication strategy, we need a value for the maximum goal cost for each performance goal that we consider as well as the communication cost. Empirically, we find these values by running the simulation repeatedly a large number of times. The maximum values for the Wumpus World and Collective Construction domains are presented in Tables 4.4 and 4.5, respectively.

A total of 162 experiments are run in two phases. In each phase, the effect of one independent variable (fitness parameter vs simulation environment) is investigated, as explained below. Fitness goal, the main independent variable in this work, is involved in both phases.

The first phase is the fitness parameter phase, and it includes 30 experiments for each domain, resulting in 60 experiments, where the fitness goal and fitness parameters are varied. Each experiment is run three times, hence a total of 180 experiments. The following command is used for both domains:

`java GA PerGoal` α β *ac* *cc*

Where:

α is the goal weight.

β is the communication weight

ac is the action cost

cc is the communication cost

This phase investigates the impact of the fitness parameters on the evolved strategies, and hence goal performance and communication cost. Specifically, this phase is trying to answer the following:

- Does increasing α guarantee better goal performance?
- Does increasing *ac* and/or *cc*, with fixed α and β values, affect the goal performance and total communication cost?

- For one parameter setting and one goal, is it guaranteed that the goal-driven communication strategy always has the best goal performance compared to strategies of other goals?

Table 4.4: Maximum values for the Wumpus World domain.

Carriers	Fighters	Wumpus	Gold	Max. Time	Max. Travel Dis.	Max. Energy	Max Eff. Dup.	Max. Comm.
5	3	6	10	1200	7503	9050	95	915
10	6	6	10	1000	11537	14410	279	2863
15	9	6	10	1000	14668	22698	555	5710
20	12	6	10	1000	19809	25959	995	11039
5	3	12	20	2000	10327	11235	108	1555
5	3	18	30	2500	12947	13362	126	2263
5	3	3	5	1000	4501	6231	50	519

Table 4.5: Maximum values for the collective construction domain.

Builders	Bulldozers	Collectors	Blocks	Debris	Structure Size	Max. Time	Max. Travel Dis.	Max. Energy	Max Eff. Dup.	Max. Comm.
3	5	5	200	100	10	1800	22316	24182	2865	3090
6	10	10	200	100	10	1200	24339	30865	2959	14692
9	15	15	200	100	10	1200	34030	39571	3756	19563
12	20	20	200	100	10	1200	42636	80534	4703	54533
3	5	5	300	150	15	2100	22580	23490	3427	6984
3	5	5	400	200	20	2400	25871	29337	3053	5904
3	5	5	500	250	25	2700	29750	32815	4185	6500

The second phase of experiments is the simulation environment phase, and it includes 24 experiments for each application domain, where each is run three times, resulting in the total of 144 experiments. In this phase, the fitness goal as well as the simulation environment are varied, while fitness parameters are fixed to unbiased values (α=0.5, β=0.5, ac=1, cc=1). The following command is used for the Wumpus World domain:

```
java GA PerGoal C F W G
```

Where:

C is the size of carrier population

F is the size of fighter population

W is the size of wumpus population

G is the amount of gold

And the following command is used for the Collective Construction domain:

```
java GA PerGoal B Bz C Bk D S
```

Where:

B is the size of builder population

Bz is the size of bulldozer population

C is the size of collector population

Bk is the number of blocks

D is the amount of debris

S is the structure size

This phase investigates the impact of the simulation characteristics on the evolved strategies, and henceforth goal performance and communication cost. Specifically, this phase is trying to answer the following:

- Does the proposed approach have a consistent performance as the scenario gets simpler and/or more complex?
- How do the evolved strategies differ when the population/task complexity change?

4.5 Parallelization of Genetic Algorithm

As mentioned previously, we utilize a steady-state Genetic Algorithm to evolve goal-driven communication strategies. Therefore, at each generation, fitness of two off-springs (two CSs) are evaluated, which involves running the simulation 20 times (ten per CS) to compute the average fitness scores. This results in 20 independent simulation runs at each GA generation, which indicates natural parallelism found in the fitness evaluation of GA evolution.

4.5.1 Hardware Configuration

We use multi-core processing to exploit the natural parallelism in GA and increase the efficiency of our computation. Our experiments run on different nodes on the Advanced Computing Facility (ACF) supercomputing cluster at the University of Kansas. Each experiment is submitted as a separate batch job. In general, jobs ran on three different types of CPUs, including E5-2660, E5-2660 v2, and E5-2670 v2, and a mix of memory (128GB, 256GB, and 512GB). However, the minimum hardware requirements are 20-core CPU and 120 GB memory for each Wumpus World job, and 12-core CPU and 90-110 GB memory for each Collective Construction job.

Due to limitation of data availability, we present hardware utilization for only the time period from May 01, 2015 to March 31, 2016, which represents mostly jobs running the Wumpus World domain, in Figures 4.1, 4.2, and 4.3. All charts are taken from KU's ACF XD Metrics on Demand (Furlani et al., nd).

Figure 4.1 depicts the average CPU hours (number of CPU cores x wall time hours) per job. According to source, the CPU usage is aggregated. For example, if a job used 1000 CPUs for one minute, it would be aggregated as 1000 CPU minutes or 16.67 CPU hours. Additionally, Figure 4.2 shows the total number for jobs running per month for the specified time period, and Figure 4.3 illustrates the wall hours per job, which is the average time, in hours, a job takes to run.

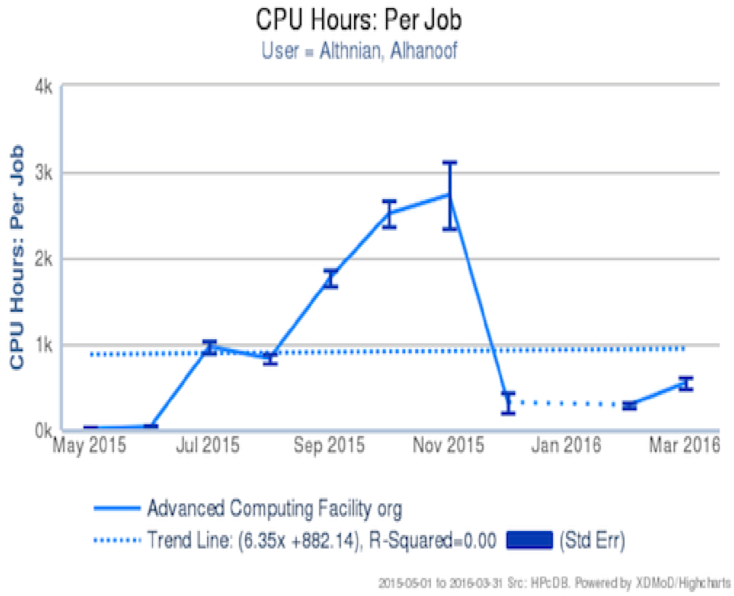


Figure 4.1: CPU hours per job.

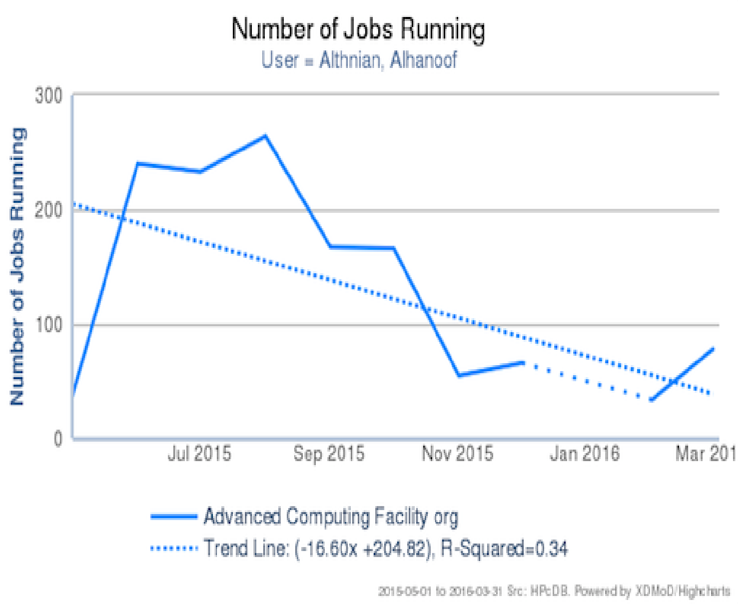


Figure 4.2: Number of jobs running.

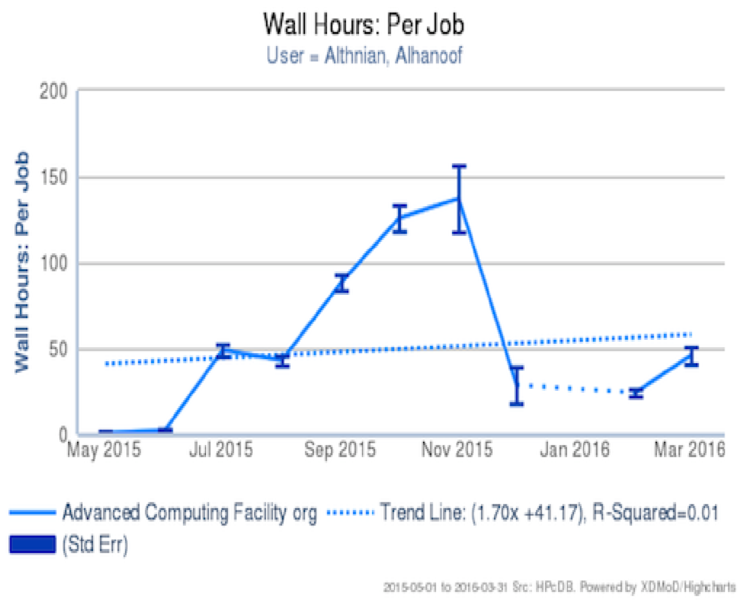


Figure 4.3: Wall hours per job.

4.5.2 Software Configuration

We use multithreading in Java to parallelize fitness evaluation in GA, by implementing `java.util.concurrent.Callable` interface. GA implementation consists of four Java classes, namely, `GA.java`, `Population.java`, `Individual.java`, and `Evolution.java`. The class, presented in Figure 4.4, is inserted in the `Individual.java`. Additionally, the code in Figure 4.6 is inserted in the function `GetFitness()` in `Individual.java` class. Another part of the evolution process that could be parallelized is evaluating the fitness of individuals in GA's initial population. For this we use Java `FixedThreadPool`. The following two attributes are added to the `Population.java` class. Also, the code in Figure 4.5 is inserted in `Population` constructor.

```
public static ExecutorService executor;
public static List<Future<String>> futures;
```

```
public class IndProcessRunner implements Callable<String>
{
      ProcessBuilder builder;
      public IndProcessRunner(ProcessBuilder builder) {
        this.builder = builder;                        }

      public String call() throws IOException, InterruptedException {
        Process process = null;
        try{process = builder.start();}
        finally{
           int errCode = process.waitFor();
               builder.redirectErrorStream(true);
               process.getInputStream().close();
               process.getOutputStream().close();
               process.getErrorStream().close();
                  }

                return "";                                             }
}
```

Figure 4.4: Callable class appended to Individual.java.

```
futures = new ArrayList<Future<String>>();
executor = Executors.newFixedThreadPool(20);
for (int i = 0; i < populationSize; i++)
{           newIndividual = new Individual(i, true);
            individuals.add(i, newIndividual);
            newIndividual.getFitness();
}
try
{          for (Future<String> future : Population.futures)
   {
        try {
               future.get();
             } catch (ExecutionException ex) {
               ex.getCause().printStackTrace();
             } catch (InterruptedException ex) {
               ex.printStackTrace();                }
      }

}
```

Figure 4.5: Code appended to the constructor of Population.java.

```java
for (int i=0;i<runs_cnt;i++)
{
    ProcessBuilder builder = new ProcessBuilder();
    builder.command("java",
         "-Xms512m",
         "-Xmx1024m",
          "-cp",".:/users/aalthnia/code/RepastRunTime/repast.simphony.runtime_2.1.0/
                bin:/users/aalthnia/code/CC/src/collectiveConst:/users/aalthnia/code/
                RepastRunTime/repast.simphony.runtime_2.1.0/lib/*:/users/aalthnia/
                code/RepastRunTime/repast.simphony.runtime_2.1.0/",
                "repast.simphony.runtime.RepastBatchMain",
          "-params","/users/aalthnia/code/CC/batch/batch_params.xml",
                   "/users/aalthnia/code/CC/CollectiveConst.rs");
    builder.directory(new
        File("/users/aalthnia/code/CCresults/id_"+GenAlg.GenA.fitfunc
       +"_a"+GenAlg.GenA.alpha+"_b"+GenAlg.GenA.beta+"_ac"+GenAlg.GenA.actcost
       +"_cc"+GenAlg.GenA.comcost+"_bl"+GenAlg.GenA.buildersCnt
       +"_bz"+GenAlg.GenA.bulldozersCnt+"_d"+GenAlg.GenA.debrisCnt
       +"_bk"+GenAlg.GenA.blocksCnt+"_s"+GenAlg.GenA.StrSize+"/Ind_"+id+"/Ins_"+i));
          builder.redirectInput(new
              File("/users/aalthnia/code/CCresults/id_"+GenAlg.GenA.fitfunc
       +"_a"+GenAlg.GenA.alpha+"_b"+GenAlg.GenA.beta+"_ac"+GenAlg.GenA.actcost
       +"_cc"+GenAlg.GenA.comcost+"/Ind_"+id+"/Ins_"+i));
    builder.inheritIO();
    IndProcessRunner pr=new IndProcessRunner(builder);
    Population.futures.add(Population.executor.submit(pr));
}
```

Figure 4.6: Code added to `GetFitness()` in `Individual.java`.

CHAPTER 5

Case Study: The Wumpus World

5.1 Scenario

A MULTI-AGENT version of the Wumpus World Problem (Russell & Norvig, 2003; Zhang, 2006) is developed using Repast Simphony (Althnian & Agah, 2015, 2016; North et al., 2005). The world consists of 12 rooms in a 140x140 grid, where one room is the drop-off room, and others either contain gold and/or wumpuses or are empty. The world contains five carriers, three fighters, six wumpuses, and ten pieces of gold, all of which are distributed randomly at the beginning of each simulation run. Carriers have the information about rooms' locations, but not their contents. Carriers are capable of finding gold and wumpuses, picking up and dropping off gold, and fighters are capable of killing wumpuses. Similar to (Zhang, 2006), the only way that fighters can know about the location of a wumpus is by receiving a message from a carrier that observed it (IS_4 in Table 5.1). Therefore, communication of this information instance is considered mandatory to enable agents to complete the task.

The information instances that agents can communicate in this domain are listed in Table 5.1. Single-value information instances include IS_1, IS_2, IS_6, and IS_7, while multi-value information

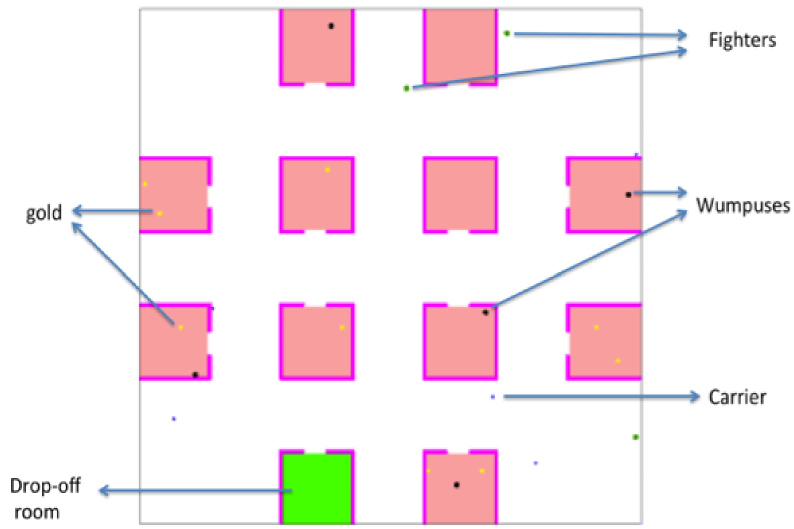


Figure 5.1: Screenshot of the Wumpus World.

instances are IS_3, IS_4, IS_5, and IS_8.

Table 5.1: Information instances of the Wumpus World and their values.

Identifier	Information Instance	Possible Values	Producer
IS_1	Carrier's Goal	ExploreRoom PickUpGold DropOffGold	Carrier
IS_2	Carrier's Goal Room	[1,12]	Carrier
IS_3	Gold Location	(x,y) 140>x,y>0	Carrier
IS_4	Wumpus Location	(x,y) 140>x,y>0	Carrier
IS_5	Empty Room	[2,12] 1 is drop-off room	Carrier
IS_6	Fighter's Goal	FindWumpus KillWumpus	Fighter
IS_7	Fighter's Goal Room	[2,12]	Fighter
IS_8	Safe Room	[2,12]	Fighter

At each time-step, every agent performs (observe, communicate, act) task cycle. In observe, agents are able to make the observations IS_3, IS_4, IS_5, IS_8, as well as decide on a goal and goal room. In the communicate step, each agent refers to the communication strategy to see if it should communicate any information at the current time-step. In the act step, all agents are able to move

UP, DOWN, RIGHT, and LEFT. In addition, carriers are able to PICK UP, and DROP OFF gold, and fighters are able to KILL a wumpus. Carriers and fighters maintain inner states, S_c and S_f, respectively, defined as:

$S_c = \{s_1, s_2\}$

$S_f = s_1$, Where: $s_1, s_2 = \{0, 1, \phi\}$.

The first element s_1 indicates agent's location, whether inside a room (1), in the hallway (0), or disregarded (ϕ). Although agents can recognize what room that they are in, this is abstracted in the state to only three cases to deliver a coherent choice to when an agent can communicate. The second element s_2 indicates the carriers' possession of gold, whether it holds gold (1), nothing (0), or disregarded (ϕ). While states keep track of agents' locations and possession, agents goals (Table 5.1) maintain what they want to achieve. Initially, all carriers have the ExploreRoom goal, and have to decide on a goal room. The choice of goal and goal room depends on the information available to each carrier about the rooms. For example, a carrier will choose the closest room that contains gold, but if it has no information about gold locations, it will choose the closest room it has never visited to explore. Needless to say, carriers avoid empty and wumpus rooms. The latter rooms can be reconsidered for exploration or gold collection if a carrier receives that the room is safe (IS_8) or contains gold (IS_3), respectively. By communicating goal and goal rooms, carriers can avoid exploring a room that is currently being explored by another carrier. In a similar manner, fighters choose the closest wumpus room, if multiple requests received, and avoid targeting a wumpus that is currently the goal of another fighter.

5.2 Experimental Results

5.2.1 Fitness Parameters

We present the experimental results for each performance goal when values of the fitness parameters are varied in Figures 5.2, 5.3, 5.4, and 5.5 (Althnian & Agah, 2016). As shown, each bar represents the goal performance for one fitness parameter setting, which has been hierarchically labeled by the goal and communication weights and further by the action and communication costs. The line chart for each cost setting represents the communication cost trend as the goal's weight increases. The performance results reported are the average of three evolution runs. Unless otherwise indicated, the evolved strategies for a single experimental setting are mostly consistent, with sometimes a slight variation in the timing strategies, such as three versus four time-steps, which does not affect the strategy's fitness and result performance. For each result, the error bar shows the standard error, computed as STDEV/SQRT (n), across n GA runs, where n=3.

The observed performance can be further justified and understood by comparing the actual evolved communication strategies (Tables 5.2, 5.3, 5.4, 5.5). Although it is enigmatic as to precisely which part of the evolved strategy caused such performance, we compare and contrast, In the forthcoming subsections, between strategies and their performance to deduce which part of the strategy, including information instances, timing, and recipient strategies, contributes to the result performance. In the tables, each row shows the strategy evolved for one parameter setting, and each column represents an information instance. Dark shaded cells represent invalid communication, or communication that only increases cost with no contribution to performance improvement. Such communication includes sending carriers goals to fighters, or vice versa, and sending gold locations to fighters. Light shaded cells convey that no communication is allowed for the corresponding information instance in the corresponding parameter setting (strategy). For clear cells, the upper line represents the recipient strategy, i.e., P2P, Bcast, or subset [2-max], and the lower line represents the timing strategy, which can take either every update (EU), in a state ($\{0,1,\phi\}$, $\{0,1,\phi\}$) for carriers (C), and ($\{0,1,\phi\}$) for fighters (F), or every time interval ([2-20] TS).

5.2.1.1 Time

Figure 5.2 shows the time performance and communication cost across different values for the fitness parameters. Firstly, in all cases of action and communication costs, the time performance of agents improves as a result of increasing its weight, which in turn costs more communication, enabled by decreasing the communication weight. ANOVA test indicates that the decrease in time is statically significant when the goal weight is increased across all action and communication costs, with ($p<0.005$). Secondly, doubling communication cost, (ac<cc), causes GA to evolve strategies with lower communication, and hence worse goal performance. This can be observed by comparing each time performance and associated communication in case of (ac<cc) with the corresponding performance and communication of other costs, i.e., bars of the same shading, in other cases (ac=cc and ac>cc).

Moreover, the time performance when (ac=cc) is almost the same as the corresponding time performance when (ac>cc). As communication cost remains the same in both cases, action cost does not affect the time fitness of the strategy. Further, it appears that low communication weight can sometimes be compensated by higher corresponding cost, which results in cases where the performance and communication is nearly similar with different goal weights. This can be observed, in Figure 5.2, when the two cases (α=β, ac=cc) and (α>β, ac<cc) are compared.

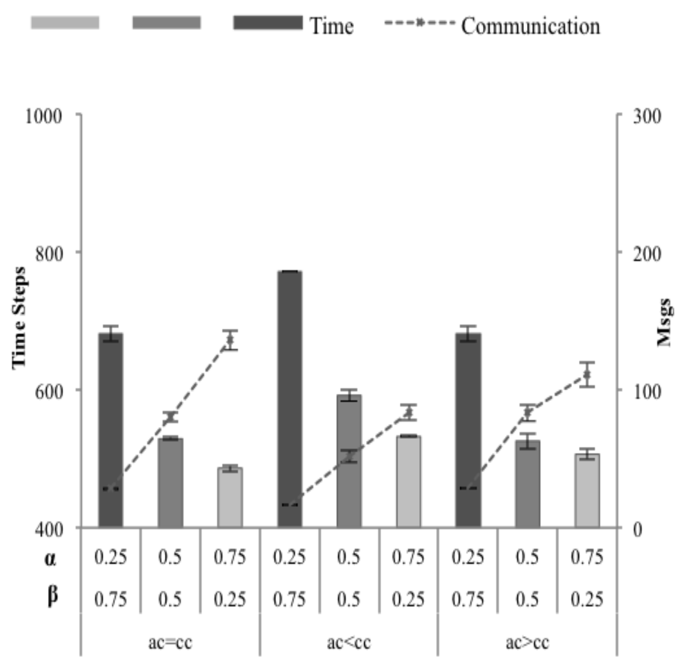


Figure 5.2: Performance of time strategies.

As can be seen in Table 5.2, the wumpus location (IS_4) is communicated to fighters in all cases, though broadcasting the information is avoided when communication cost doubles and when communication weight is higher, in efforts to minimize communication. Apparently, for one cost setting, the communicated information instances increase as a consequence of decreasing communications' weight (β). Also, the evolved strategies of one weight setting have lower number of communicated information instances when the communication cost is high, compared to the other two cost settings. For example, when ($\alpha<\beta$), agents can communicate IS_4 and IS_8, both when (ac=cc) and (ac>cc), while only IS_4 is allowed to be communicated when (ac<cc).

Table 5.2: The evolved time communication strategies.

Fitness Parameters			Information Instances (IS_i)							
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
ac=cc	$\alpha<\beta$	C								4 (2TS)
		F				2 (10TS)				
	$\alpha=\beta$	C			2 (3TS)		Bcast (EU)			P2P (9TS)
		F				Bcast (5TS)				P2P (14TS)
	$\alpha>\beta$	C		P2P (5TS)	2 (1,0)		Bcast (EU)			4 (17TS)
		F				Bcast (9TS)				
ac<cc	$\alpha<\beta$	C								
		F				2 (EU)				
	$\alpha=\beta$	C					3 (1,1)			P2P (19TS)
		F				2 (2TS)				
	$\alpha>\beta$	C			3 (5TS)		Bcast (EU)			P2P (1)
		F				2 (3TS)				
ac>cc	$\alpha<\beta$	C								2 (4TS)
		F				2 (10TS)				
	$\alpha=\beta$	C			2 (2TS)		Bcast (EU)			P2P (1)
		F				Bcast (15TS)				P2P (1)
	$\alpha>\beta$	C		P2P (5TS)	2 (1, 0)		Bcast (EU)			4 (17TS)
		F				Bcast (9TS)				

Moreover, the value of each information instance to time performance could be evaluated by investigating the circumstances (or parameter settings) under which each information instance is allowed to be communicated. For example, mandatory information, such as IS_4, is communicated to fighters in all parameter settings, although with different strategies. On the contrary, communicating IS_1 and IS_4 to carriers, and IS_5, IS_6 and IS_7 to fighters is not allowed in any setting, hence we can conclude that these information has no contribution to minimizing time. However, we still cannot confirm whether they are neutral or unfavorable to time performance. The second most important information to time appears to be the safe room (IS_8) to carriers, as it is avoided in only a single case that is when both communication weight and cost are high ($\alpha<\beta$, ac<cc). The prominence of this information comes from the fact that it draws carriers attentions to formerly dangerous rooms, which have been avoided by all carriers that observed the wumpus. The contribution of this information in minimizing time is noticeable by comparing the time performance of the strategies when ($\alpha<\beta$, ac<cc) and ($\alpha<\beta$, ac>cc) or ($\alpha<\beta$, ac=cc), which has been confirmed by t test (p=0.0007). It seems that communicating carriers goal room (IS_2) can decrease time, but appears to cost high communication as carriers update their goal rooms frequently. Therefore, it is only communicated when time outweighs communication and further communication cost is not high. It appears that the next influential information is the empty room (IS_5), followed by the gold location (IS_3). The former is dropped from three strategies, when communication outweighs time, whilst the latter seems to be either more expensive or less influential as it is dropped from four strategies; three of them when communication outweighs time and one when weights are equal but communication cost is higher. The later parameter setting is the only case when IS_5 is communicated to only three carries if the sender is in a room and holding gold, as opposed to broadcasting every update in all other settings.

Another observation is that carriers' goal (IS_1) is not communicated in all strategies. Since initially all carriers have the ExploreRoom goal and each carrier knows that others have this goal, not communicating any update of the goal will make each carrier believe that others always have ExploreRoom goal. Since carriers do not explore a room that is currently explored by others, hence if carriers' goal room (IS_2) is communicated, then carriers will not explore a room that is currently explored or even visited by another carrier to pick up gold. This behavior seems to work better in minimizing time than the originally implemented behavior, and thus has been enforced by the evolved strategies.

5.2.1.2 Travel Distance

Mutual observations can be made for the performance of travel communication strategies in Figure 5.3. For instance, the distance traveled by agents decreases as a result of increasing the travel distance weight. However, for one cost setting, increasing the goal weight does not necessarily result in significant improvement in the goal performance. This can be seen, in Figure 5.3, when the goal weight increases from 0.25 to 0.5, with (ac>cc), as (p=0.13). This can occur if any further significant improvement in goal performance would require a significant increase in communication that, given the current weight setting, would result in worse fitness. The steep increase in communication when goal's weight increases to 0.75, for the same cost setting, confirms this. Moreover, similar performance is obtained with two different fitness parameter settings: (1) (α>β, ac<cc), and (2) (α=β, ac=cc), with (p=0.47), which can be verified by the close evolved strategies in Table 5.3. Although the difference in weights, the variations of the action and communication costs in the two cases even out the importance of the two costs, resulting in a similar strategies and performance. The strategy (α=β, ac>cc) shares a close performance and strategy with the two previous ones for the same aforementioned reason, i.e., any further significant improvement in goal performance over the strategy (α<β, ac>cc) would require a significant increase in communication. Unlike the time fitness, increasing the action cost does impact the travel fitness, and hence the evolved strategies, since action cost includes the cost of travelling. By comparing the system's performance in the three cases of cost settings, it can be seen that, on one hand, the evolved strategies, when (ac<cc), maintains the minimum amount of communication, and hence maximum travel distance, comparing to its counterparts in (ac=cc and ac>cc). On the other hand, strategies evolved with (ac>cc) tend to communicate more and travel shorter distance, compared to its counterparts in (ac=cc and ac<cc). Moreover, we notice that, except for the high communication cost setting, the evolved strategies have higher communication standard error when the communication weight is

low (β=0.25), compared to other weight settings. The reason for this is that when communication is given relatively low weight (and cost), compared to goal, as in ($\alpha>\beta$, ac>cc), GA may sometime exaggerate in communication, in efforts to further minimize the cost of travel. Therefore, any extra communication evolved, represented in more recipients or too often communication, is actually neutral that increases communication, but does not negatively affect the goal performance, verified by the low goal's standard error.

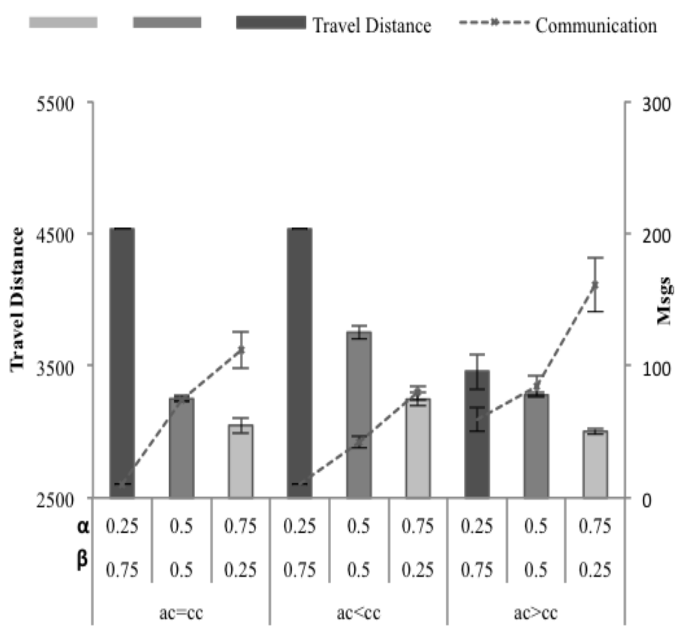


Figure 5.3: Performance of travel distance strategies.

Table 5.3 compares the travel communication strategies. Unlike the time strategies, broadcasting the wumpus information (IS_4) is avoided in all parameter settings since it triggers response from all fighters, and hence increases the total travel distance. Moreover, IS_3 is avoided in all but one strategy, when ($\alpha>\beta$, ac>cc), due to over communication. Therefore, communicating IS_3 does not contribute to minimizing travel distance. Rather, IS_3 may, in some scenarios, attract carriers to the gold locations, which may result in carriers ignoring nearby rooms and travelling longer distance to collect gold. IS_8 seems to be less important to travel than time, as it is dropped from all strategies where communication cost is higher and when it costs as much as action but has higher weight.

Table 5.3: The evolved travel distance communication strategies.

Fitness Parameters			Information Instances (ISi)							
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
ac=cc	$\alpha<\beta$	C								
		F				P2P (EU)				
	$\alpha=\beta$	C				P2P (20TS)	Bcast (EU)			P2P (11TS)
		F				2 (1,0)				
	$\alpha>\beta$	C		P2P (1,0)		2 (EU)	Bcast (20TS)			P2P (15TS)
		F				P2P (EU)				
ac<cc	$\alpha<\beta$	C								
		F				P2P (EU)				
	$\alpha=\beta$	C					3 (1,1)			
		F				P2P (EU)				
	$\alpha>\beta$	C				P2P (13TS)	Bcast (EU)			P2P (1)
		F				2 (4TS)				
ac>cc	$\alpha<\beta$	C					Bcast (EU)			P2P (12TS)
		F				P2P (EU)				
	$\alpha=\beta$	C				P2P (1, ϕ)	Bcast (16TS)			P2P (15TS)
		F				P2P (EU)				
	$\alpha>\beta$	C		P2P (8TS)	2 (7 TS)	P2P (17 TS)	Bcast (EU)			P2P (17TS)
		F				2 (1, ϕ)				P2P (6TS)

Similar to time, carriers' goal rooms (IS_2) is allowed to be communicated in only two strategies, when goal weight is high and communication cost is not high. Also, IS_5, IS_6, and IS_7 are never communicated to fighters, and IS_1 is never communicated to carriers for the same reason explained previously. IS_5 appears to be as important to travel as it is to time. However, since doubling action cost affects travel fitness as oppose to time fitness, IS_5 is communicated when ($\alpha<\beta$, ac>cc) as well. This information can help carriers avoid empty rooms, hence focus more on exploring rooms with potential gold. Moreover, unlike time, travel distance allows communicating IS_4 to carriers, and similar to gold locations (IS_3) to time, this information instance is avoided only when communication outweighs travel distance and when their weights are equal but communication cost is higher. IS_4 is communicated to the closest single or two carriers, hence the number of working carriers continually decreases as the task progresses, which results in shorter travel distance. However, broadcasting or communicating IS_4 to multiple carriers is avoided, as doing so causes most carriers to reach idle state early, which leaves more work to only one or a few carrier(s), and hence increase total travel distance. The safe room (IS_8) appears to be less important to travel distance than it is to time, since it is dropped from three travel strategies, as oppose to one time strategy. Further, when IS_8 is communicated, it is only sent to the closest carrier (P2P), while in some time strategies, IS_8 is communicated to two and four carriers. The gold location (IS_3) is only communicated in one strategy, when ($\alpha>\beta$, ac>cc), and to the closest two carriers. We notice that travel strategies are very considerate to sharing potential gold locations to carriers (by communicating IS_3 and IS_8). It is believed that communicating such information to remote carriers can increase travel distance, as those carriers are distracted from nearby rooms and encouraged to travel to farther rooms that contain gold or just became safe. Similar to time, carriers' goal (IS_1) is not communicated in all strategies for the same aforementioned reason.

5.2.1.3 Energy

As previously stated, energy is a single-objective performance goal; hence only costs variation is applicable. Figure 5.4 shows the energy performance of the evolved strategies, presented in

Table 5.4, when action and communication costs are varied. We observe that, on one hand, when communication cost is doubled (ac<cc), communication is significantly reduced, compared to the case when communication cost is low (ac=cc), with (p=0.004), though the total amount of energy consumed is significantly higher when (ac<cc), with (p=0.001). On the other hand, communication is significantly increased when (ac<cc), comparing to (ac=cc) with (p=0.001), as a consequence to doubling the action cost, yet no difference in the amount of energy consumed, between the two cost settings, is obtained, (p=0.25). Therefore, we believe that any extra communication, evolved in the strategy of the cost setting (ac>cc), is not indeed necessary, as the strategy for the setting (ac=cc) can perform as good as the former strategy, yet with significantly less communication.

Energy consumption can be minimized by traveling shorter distance and/or communicating fewer messages. Unlike other (multi-objective) performance goals, the absence of goal and communication weights can yield GA to accept communication strategies with any communication cost as long as it consumes less energy. Therefore, it may not be clear as to why a specific information instance is communicated or avoided, and hence comparing the evolved energy strategies in different cost settings is ineffectual.

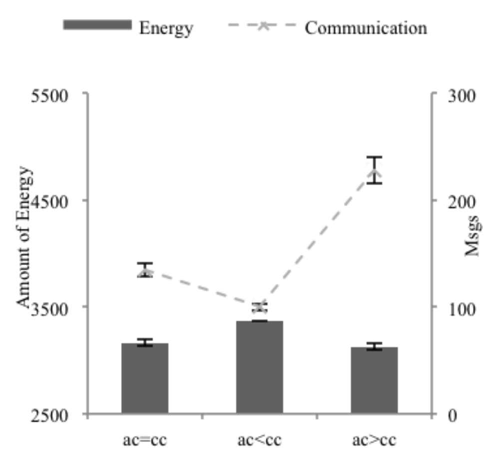


Figure 5.4: Performance of energy strategies.

Table 5.4: The evolved Energy communication strategies.

Fitness Parameters		Information Instances (IS_i)							
Costs		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
ac=cc	C		P2P (10TS)		3 (7 TS)	Bcast (1, 0)			2 (1)
	F				P2P (EU)				
ac<cc	C				P2P (10TS)	Bcast (1, Phi)			P2P (EU)
	F				2 (2 TS)				2 (2 TS)
ac>cc	C	P2P (7 TS)	P2P (10 TS)	3 (4 TS)	2 (4 TS)	Bcast (1, 0)			P2P (5 TS)
	F				2 (2 TS)				2 (5 TS)

5.2.1.4 Effort Duplication

As the main aim of this performance goal is to prevent two agents from targeting the same goal, we notice the popularity of P2P as the recipient strategy in all parameter settings. The purpose is that no two agents are told about the same information to avoid triggering them to the same goal. We observe close performance, in Figure 5.5, for the corresponding strategies when (ac=cc) and (ac>cc). Yet, an interesting observation can be made by comparing the strategies evolved for the two settings ($\alpha > \beta$, ac=cc) and ($\alpha > \beta$, ac>cc). Due to the recipient strategy evolved for communicating IS_4 to fighters in the latter setting (2 versus P2P), GA further evolved communicating IS_6, IS_7, and IS_8 to fighters. Communicating IS_8 to fighters is always paired with communicating the wumpus location (IS_4) to a subset of 2, rather than P2P. As one might imagine, sending a wumpus location to two fighters increases the likelihood of targeting the same wumpus. Therefore, communicating the news that the wumpus has been killed, i.e., safe room (IS_8), can reduce the occurrence of this situation. Due to the low communication weight, GA has additionally evolved communicating the fighters goals (IS_6) and goal rooms (IS_7), to further prevent targeting one wumpus by multiple fighters. T test suggests that the two strategies have close effort duplication fitness (p=0.18), due to the low weight, and hence contribution, of communication cost to the fitness.

We observe (Figure 5.5) that the performance is close when goal's weight increases from 0.25 to 0.5, when (ac<cc), for the same reason mentioned previously. Once more, any further reduction

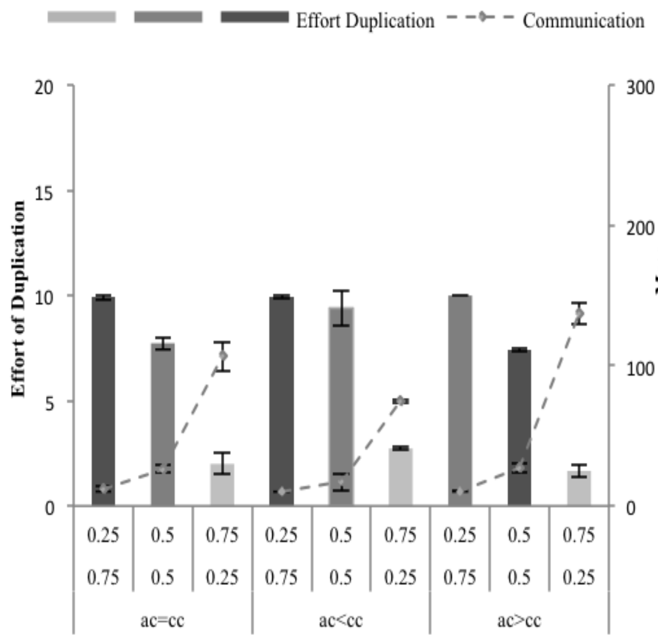


Figure 5.5: Performance of effort duplication strategies.

in duplication of effort, requires a significant increase in communication, which, given the cost setting, results in worse fitness. This can be verified by the significant increase in communication when goal's weight increases to 0.75. However, strategies evolved at the weight setting (α=β, ac<cc) are more variant that those evolved at (α<β, ac<cc). The reason is that when both communication weight and communication cost are high, GA ensures that minimal communication is the best strategy. However, when communication weight is equalized with goal weight, GA has once, out of three runs, evolved a strategy with higher communication than minimal communication in effort to improve goal performance. The evolved strategy is similar to that at (α=β, ac=cc), which has worse fitness in this parameter setting than minimal communication.

The evolved effort duplication strategies (Table 5.5) suggest that IS_2 is communicated in three strategies, where goal outweighs communication. This confirms our previous observation that communicating IS_2 is costly, hence it was communicated, in time and travel distance strategies, only when goal outweighs communication and communication weight is not high. In this performance goal, however, IS_2 is allowed to be communicated when communication cost is high, due to the crucial role it plays in reducing effort duplication. Moreover, IS_2 is communicated more often

Table 5.5: The evolved effort duplication communication strategies.

Fitness Parameters			Information Instances (ISi)							
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
ac=cc	$\alpha<\beta$	C								
		F				P2P (EU)				
	$\alpha=\beta$	C								P2P (19 TS)
		F				2 (7 TS)				P2P (2 TS)
	$\alpha>\beta$	C		P2P (3 TS)			Pcast (1, ϕ)			P2P (13 TS)
		F				P2P (EU)				
ac<cc	$\alpha<\beta$	C								
		F				P2P (EU)				
	$\alpha=\beta$	C								
		F				P2P (EU)				
	$\alpha>\beta$	C		P2P (EU)						P2P (14 TS)
		F				P2P (EU)				
ac>cc	$\alpha<\beta$	C								
		F				P2P (EU)				
	$\alpha=\beta$	C								P2P (19TS)
		F				2 (7 TS)				P2P (1, ϕ)
	$\alpha>\beta$	C		P2P (2 TS)			P2P (2 TS)			P2P (12 TS)
		F				2 (9 TS)		P2P (0, ϕ)	P2P (0, ϕ)	2 (12 TS)

than it is in the previous two goals. Communicating empty rooms (IS_5) seems to be less important and hence is only communicated in two strategies. It is observed that gold locations (IS_3) are not communicated at all because sharing gold locations causes more carriers to target the same gold, which increases duplication of effort. Moreover, communicating the wumpus location (IS_4) to carriers is avoided in all strategies, as it results in carriers avoiding all wumpus rooms and hence left with few room choices, which increases simultaneous rooms exploration. Communication of the safe rooms (IS_8) to carriers seems to be important in lowering effort duplication, as it is avoided only when communication outweighs goal performance and in cases where communication has equal weight to that of performance, but with high cost.

5.2.2 Fitness Goal

In this section we examine the strategies evolved from a different angle. We consider each case of fitness parameters, and study the impact of changing the fitness goal on the evolved strategies and goal performance of the system. This is done by considering each case of weight setting separately and comparing the fitness and performance of communication strategies of various goals with respect to each performance goal. Each of the Figures in this section include the fitness and performance of the minimum communication strategy, which communicates every update of the wumpus location (IS_4) as P2P to fighters, as evolved by GA when the fitness function was set to only the communication cost. This enables the comparison between the evolved strategies and the minimum communication strategy.

5.2.2.1 $\alpha > \beta$

With this weight setting, the goal performance is given the most weight of the fitness. Therefore, GA can increase communication. A small improvement in the goal performance, attained by a greater increase in communication is accepted, as a consequence of the relative weight of goal performance versus communication.

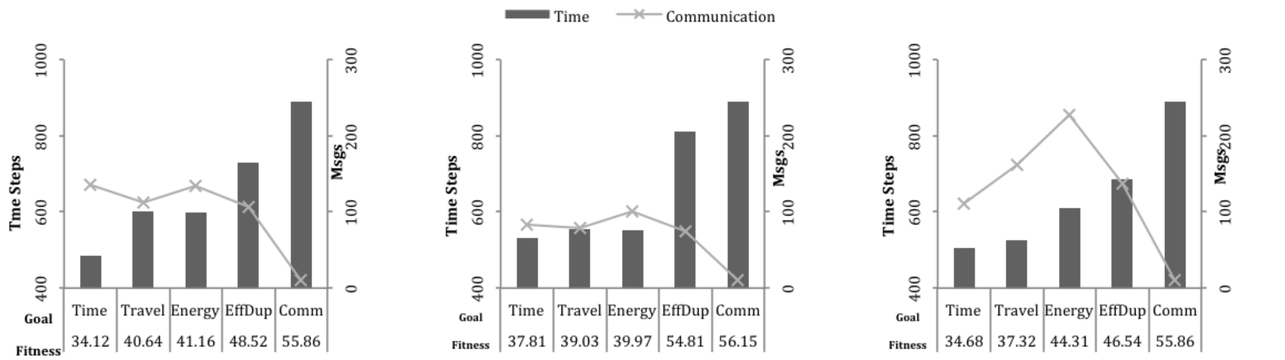
Figure 5.6 compares the time performance of the time communication strategy with strategies of other goals. In the Wumpus World domain, minimization of time and minimization of travel distance are not always conflicting goals. If the goal is to minimize travel distance, one would suppose that GA should evolve communication what leads agents to collecting gold and avoiding empty and wumpus rooms, and thus every move is worthwhile. This behavior may in some cases, concurrently, reduce the time needed to complete the task, as agents waste no time on actions that make no progress in the assigned task.

That being the case, we observe a close, yet inequivalent, time performance (p=0.28) and fitness (p=0.27) for the two strategies in Figure 5.6b. However, when the communication cost is not high (Figures 5.6a and 5.6c), minimizing travel distance does not yield fast performance for the system. In fact, it results in a significantly higher time (p=0.02 and p=0.02) and fitness (p=0.01 and p=0.04) when (ac=cc and ac>cc), respectively.

Similarly, minimizing time achieves a comparable travel distance to that of travel strategy (p=0.09), when communication cost is high (ac<cc), as shown in Figure 5.6b. Yet, the travel strategy performs significantly better in the other two parameter settings, (p=0.01) when (ac=cc), and (p=0.02) when (ac>cc). As travel strategy communicates less in Figure 5.7a, it achieves a significantly better fitness (p=0.0003) than the time strategy, whilst the high communication in Figure 5.6c yields no significant travel fitness than that of the time strategy, (p=0.12). To understand why, we shall examine the evolved strategies for the two goals, given in Tables 5.2 and 5.3 in the previous section.

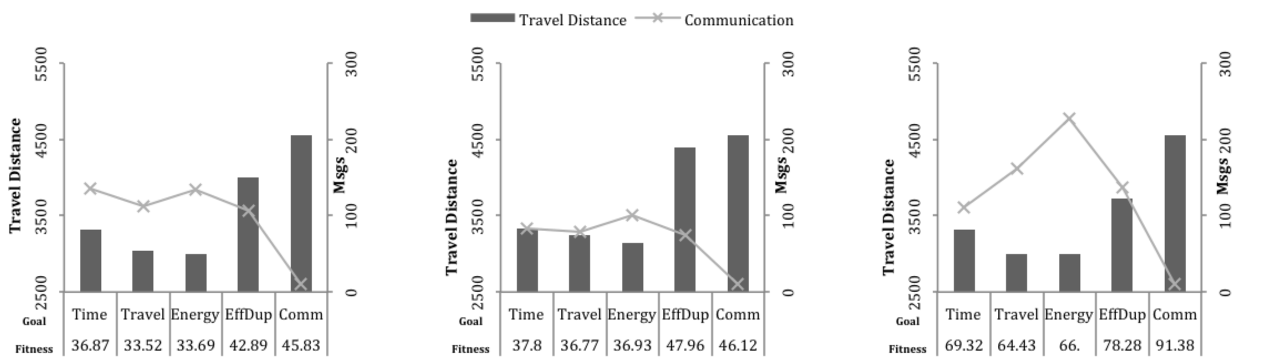
When action and communication costs are equal, key differences in the evolved time and travel strategies exist that are believed to achieve the significant performance. First, time strategy broadcasts the wumpus locations to fighters and allows communicating the safe room (IS_8) to four closest carriers as well as gold locations to the two closest carriers. These together guarantee quick collection of gold that exist in wumpus rooms. In the travel strategy, however, only the closest fighter is informed about the wumpus location, hence one fighter will target the wumpus, which may take longer time as the fighter could be responding to another request, but definitely costs less travel distance as only one fighter is traveling, rather than three. In addition, only the closest carrier is informed when a wumpus is killed. Therefore, only the closest carrier to the room considers exploring the room and yet never shares gold locations with others, if any is found. As explained before, travel strategies are cautious about sharing potential gold locations among carriers, by communicating IS_3 and IS_8, as it may cause longer travel distance. Moreover, the travel strategy communicates the wumpus locations to not only fighters, but also to the closest two carriers. This means that these carriers will avoid the wumpus room, which limits their room choices, and hence soon come to a standstill. Therefore, the number of working carriers will continually decrease as the task progress, which results in lower total travel distance and longer time to complete the task, comparing to the time strategy. Second, both time and travel strategies allow broadcasting empty rooms (IS_5) to carriers, although time strategy allows communicating it right away, while with the travel strategy, carriers have to wait 20 time-steps to communicate it. Moreover, both strategies communicate the carrier goal room (IS_2) to the closest carrier. These two information instances can help carriers not to waste time travelling to rooms that cannot make any progress in the assigned task or rooms that are already targeted by another neighbor carrier. With the effort duplication strategy, only the closest carrier is told that a room is empty. Broadcasting such information is avoided in this case because it limits carriers'room choices, and thus increases the possibility that multiple carriers target an unempty room. This strategy for sharing IS_5 along with disallowing communication of IS_4 to carriers cause a remarkably higher travel distance, compared to the travel strategy ($p=0.0002$), while this strategy combined with communicating IS_8 as P2P cause the significant longer time to finish the task, comparing to the time strategy ($p=0.0001$).

When communication cost is high, time and travel strategies share equivalent strategies for communicating IS_4, IS_5, and IS_8. The difference between the two strategies is that time strategy communicates the gold locations IS_3, while travel strategy communicates IS_4 to carriers. As mentioned previously, this information is beneficial to improving the corresponding goal performance. However, it seems that the current strategies used for sharing the two information instances, for example, only the closest carrier is informed about the wumpus location in the travel strategy, do not make such a significant difference in the corresponding performance goal. Effort duplication



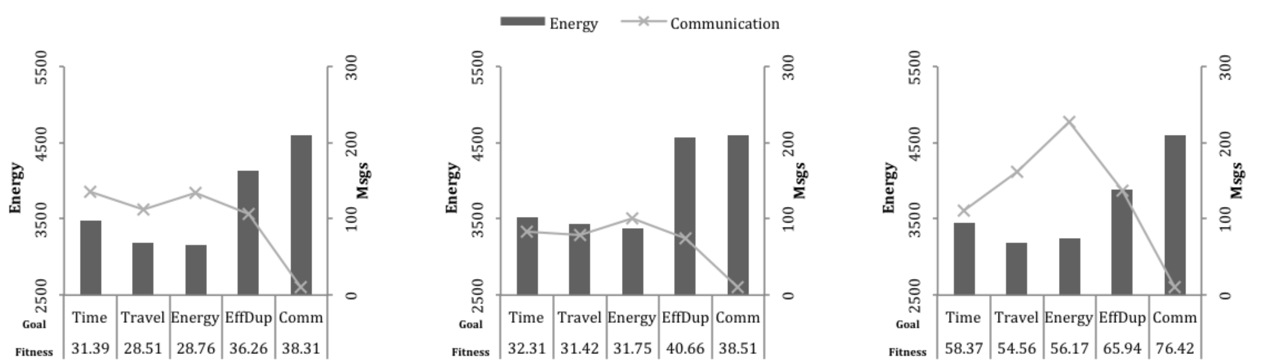
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.6: Time performance of communication strategies of various goals ($\alpha>\beta$)



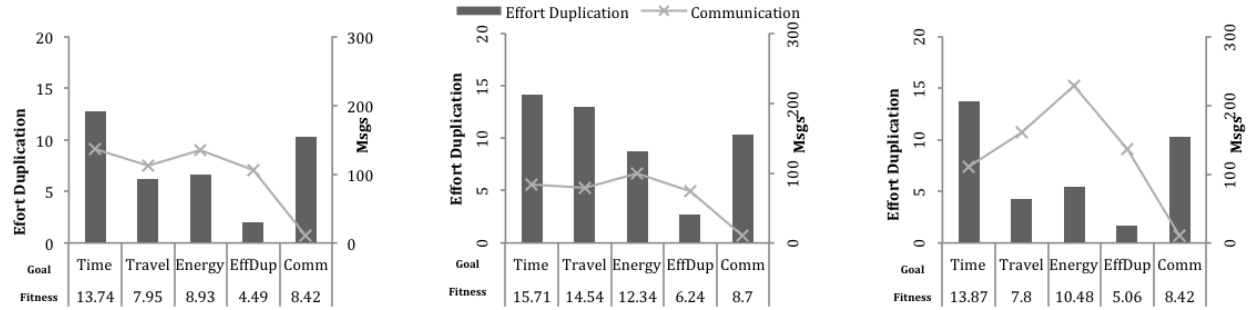
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.7: Travel performance of communication strategies of various goals ($\alpha>\beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.8: Energy performance of communication strategies of various goals ($\alpha>\beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.9: Effort duplication performance of communication strategies of various goals ($\alpha > \beta$)

strategy has further disallowed communicating IS_5, hence it costs significantly higher in terms of time and travel distance.

When the action cost is high, time strategy maintains a similar performance as well as strategy to the cost setting (ac=cc), because the action cost has no effect on the time fitness. The travel strategy, however, increases communication to reduce action (i.e. travel) cost. As shown in Table 5.3, the travel strategy allows communicating the wumpus locations to two fighters, rather than one. The analysis revealed that the two recipient strategies for communicating IS_4 to fighters (P2P vs 2) costs less travel distance in different scenarios, depending on the locations of fighters and wumpuses. For example, if wumpuses are located in nearby or the same room(s), then P2P would cost less in travel distance, while the other strategy costs less if they are apart. Yet, the difference is not significant. Though to guarantee that two fighters do not target the same wumpus, the strategy allows sending the safe room to the closest fighter, hence making every step worthwhile. In addition, the strategy communicates every update of IS_5, rather than every 20 time-steps, as when (ac=cc), and also allows sending the gold location (IS_3). This strategy costs significantly less in travel distance than the time strategy, but due to the significant higher communication cost, no significant improvement in travel fitness is achieved over the time strategy. The effort duplication has lower time and travel distance, compared to the previous two cost settings. The reason is that the strategy allows communicating IS_4 to two fighters, rather than one, and to ensure that duplication of effort is minimized; it further communicates fighters' goal and goal rooms, as well as information about killed wumpuses to fighters.

Energy performance of the strategies is depicted in Figure 5.8. As can be seen, the strategies' relative performance resemble their corresponding's travel performance. The reason is that cost of traveling amounts to most of the energy consumed. As stated previously, agents consume energy when they perform actions, such as move, pick up, drop, and shoot, and when they communicate. Although energy strategy has a close performance to the travel strategy with respect to both travel distance and energy, it does so with a higher amount of communication. Since communication cost makes up only a quarter of the total strategy's fitness, we observe a slight increase in the fitness of the energy strategy comparing to the travel strategy's fitness. Recall that energy is a single-objective goal and travel distance is a multi-objective goal, we conclude that assigning weights to the goal and communication costs can lead to better balance between the two costs. As time and effort duplication fitness do not take into account action cost, we observe the high energy consumed by the two strategies when action cost doubles, comparing to travel and energy strategies.

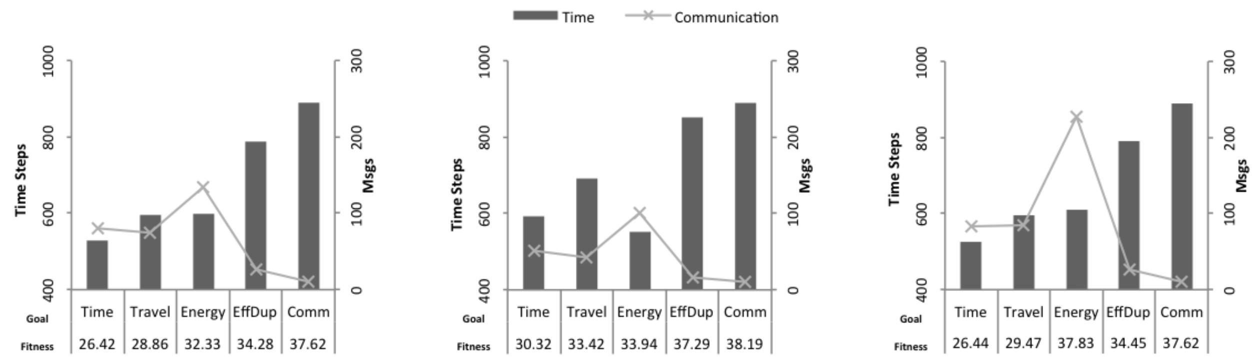
Figure 5.9 presents the strategies performance with respect to effort duplication. The effort duplication strategy outperforms all other strategies with the best goal performance and fitness. We observe that time strategy has the maximum duplication of effort, compared to other strategies, in both (ac=cc) and (ac>cc). We suggest that the first reason is due to broadcasting the wumpus location to all fighters as oppose to one and two fighters in the other strategies, and yet not communicating IS_8 to fighters as in other strategies when IS_4 is communicated to multiple fighters. Second, the time strategy communicates IS_8 to four carriers, compared to one in the other two strategies. When communication cost is high, effort duplication of time strategy increases slightly, while for travel strategy, it increases significantly to be close to that of the time strategy. The reason for the increase in effort duplication is disallowing communication of carriers' goals (IS_2) in both strategies. Yet, the increase for the time strategy is not significant as IS_4 is not broadcast, compared to other cost settings. While for the travel strategy, it communicates IS_4 to more fighters, comparing to other cost settings, and further does not communicate IS_8 to fighters, hence the close effort duplication performance for the two strategies. In all cost settings, minimum communication strategy has high effort duplication, although no effort duplication happens among fighters as only one fighter is informed about each wumpus location. The reason is due to the lack of communication, and hence coordination, among carriers.

5.2.2.2 α=β

In this case, goal and communication costs contribute equally to the total strategy's fitness. As a result, GA approves increasing communication cost only if it improves goal performance for the same, or close, proportion. Figure 5.10 depicts the time performance of the strategies. Unlike the previous weight setting, we observe disparity between the time performance of the time and travel strategies when communication cost is high (p=0.04), and close time performance when action and communication costs are equal (p=0.07) and when action cost is high (p=0.11). However, no significant travel performance is observed for the travel strategy over the time strategy in all cost settings, (p=0.11, 0.06, 0.25), for Figures 5.11a, 5.11b, 5.11c, respectively.

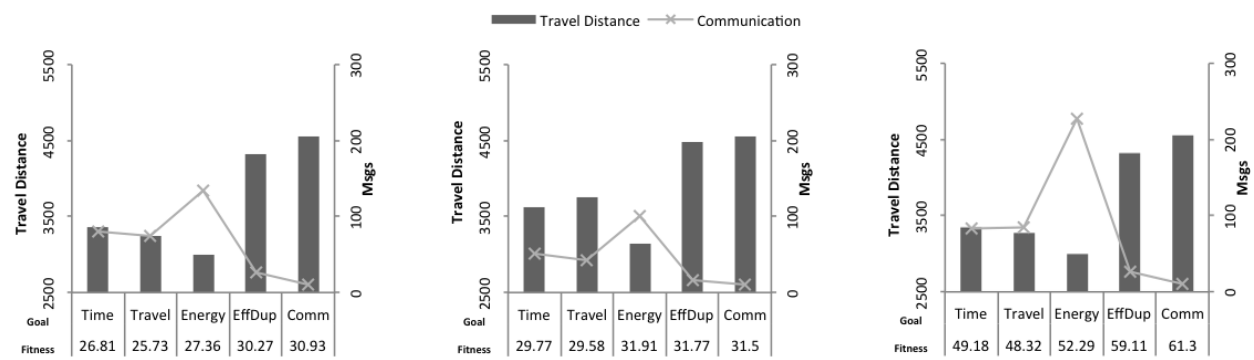
A comparison of time and travel strategies in Tables 5.2 and 5.3, when (ac<cc), highlights the importance of communicating IS_8 to carriers in minimizing the time needed to complete the task, along with sending IS_4, 'wumpus location', to two fighters rather than only one fighter. Since the difference between the two strategies serves only the time performance, as the difference between IS_4 recipient strategies that affects travel performance is not significant, we observe close travel performance between the time and travel strategies, in Figure 5.11b. The energy strategy allows agents to complete the task in significantly less time than the time strategy does, (p=0.017), due to broadcasting IS_5 to carriers and communicating IS_8 to two fighters. Communicating the latter information instance and IS_4 to a carrier contribute to significantly lowering the distance traveled by agents than that of the travel strategy (p=0.0001), as well as that of the time strategy (p=0.0002). Both effort duplication and minimum communication strategies communicate minimally; hence the notable longer time to complete the task.

In the two cost settings (ac=cc) and (ac>cc), no significant performance differences are obtained between the time and travel strategies with respect to both time and travel distance. The dif-



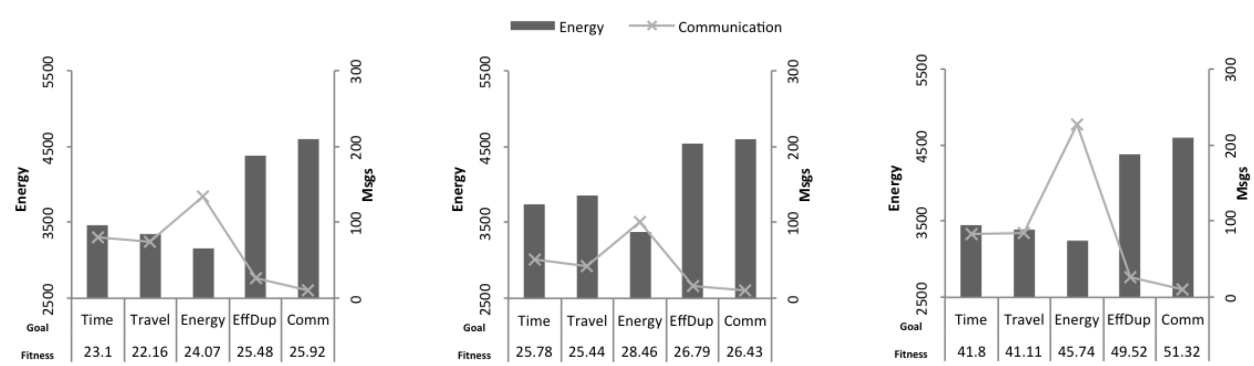
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.10: Time performance of communication strategies of various goals (α=β)



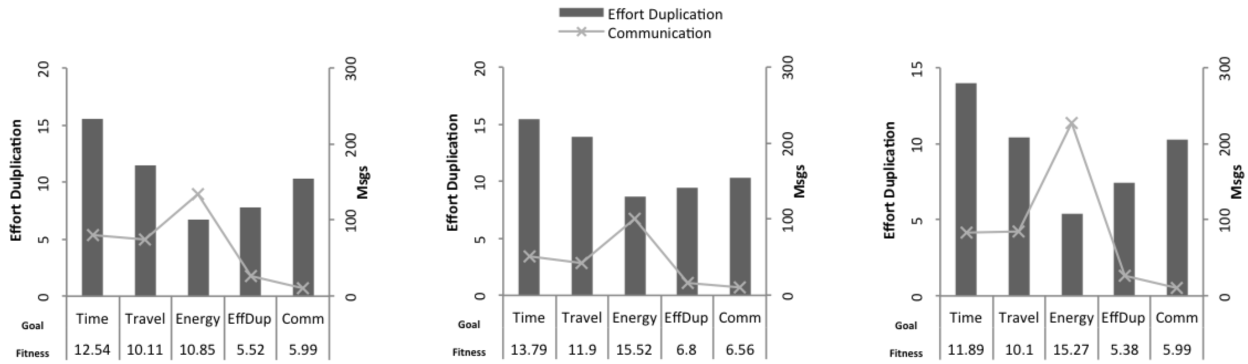
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.11: Travel performance of communication strategies of various goals (α=β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.12: Energy performance of communication strategies of various goals (α=β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.13: Effort duplication performance of communication strategies of various goals (α=β)

ference between the evolved strategies resembles that of the previous weight setting when (ac<cc). In both cost settings, the difference between the two strategies is that time communicates the gold locations (IS_3), while travel strategy communicates IS_4 to a carrier. In addition, the time strategy broadcasts IS_4 to fighters but minimizes their travel distance by communicating IS_8 to fighters, while the travel strategy communicate IS_4 to the closest one or couple of fighters. The effort duplication strategy takes significantly longer time to complete the task, comparing to the time strategy, due to not communicating empty rooms (IS_5). Yet, it takes shorter time than the minimum communication strategy, since it communicates IS_8.

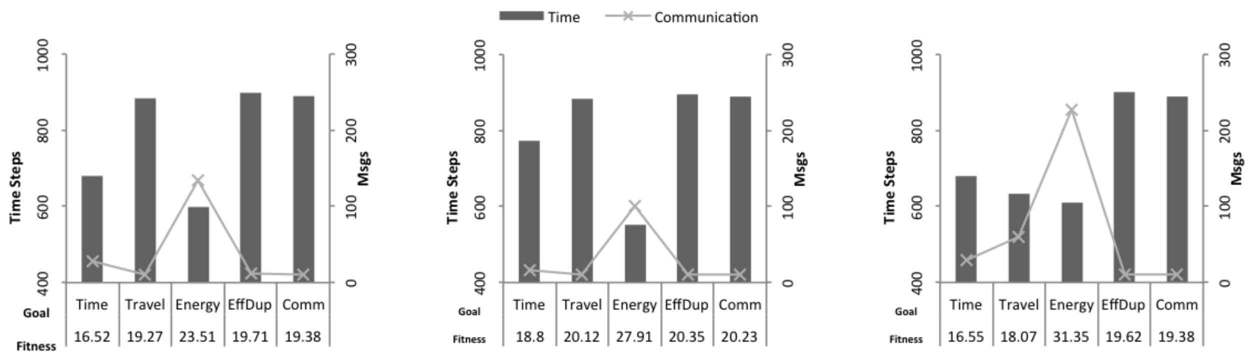
The relative energy performance of the strategies is shown in Figure 5.12, and it is similar to that of travel performance for the aforementioned reasons. We observe that although energy strategy has lower amount of energy consumed than that of the travel strategy, it is surpassed by the latter with respect to energy fitness, due to the high communication. Figure 5.13 shows the effort duplication performance for the strategies. The time strategy has the worst performance in all cost settings, with similar performance for the travel strategy when (ac<cc), due to not communicating IS_8 to carriers. The negative effect of sharing IS_5 to many carriers can be observed by comparing the travel and effort duplication strategies when communication cost is high.

5.2.2.3 $\alpha<\beta$

Communication cost outweighs the goal cost in this setting. The system designer is usually very concerned about communication cost, but still interested in seeing whether low communication can improve the system's performance. Thus, the aim is to improve the goal performance, but with minimum cost of communication.

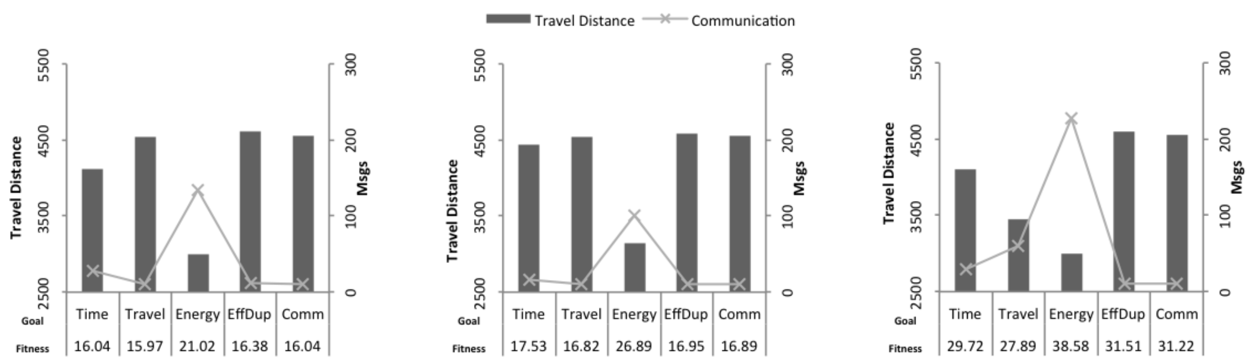
Figure 5.14 shows the time performance of the strategies. Aside from energy, the time strategy outperforms all other strategies, (Figure 5.14a), as they communicate minimally. The time performance of the strategies become closer, in Figure 5.14b, as the time strategy drops IS_8, while all others maintain minimum communication. In Figure 5.14c, the travel strategy completes the task almost as fast as the time strategy (p=0.06), as it increases communication over other cost settings, in efforts to minimize the high travel cost. While the time strategy has the advantage of communicating wumpus locations to two fighters and safe rooms to two carriers, rather than a single recipient in the travel strategy, the latter has the advantage of broadcasting empty rooms to carriers. The pros and cons in both strategies result in close time performance. However, because of the high communication in travel strategy comparing to time strategy, the latter has significantly better time fitness (p=0.04). Since the energy strategy was evolved with a single-objective fitness function, it is not affected by varying the fitness weights. Therefore, in both (ac=cc) and (ac>cc) cost settings, we observe insignificant lower time consumed by this strategy comparing to the time strategy, (p=0.07 and p=0.17, respectively). Yet, its time fitness is significantly worse than the time strategy (p=0.00006, p=0.0009, respectively), due to the high amount of communication. In the (ac<cc) cost setting, however, the energy strategy consumes significantly less time than the time strategy to complete the task, (p=0.00001), but again time fitness of the time strategy is significantly better than that of the energy strategy (p=0.0007).

In Figure 5.15a, although time strategy has lower travel distance than the travel strategy (p=0.01), due to high communication, no significant difference is found between the travel fitness of the two strategies (p=0.30). The two evolved strategies are close when the communication cost doubles (ac<cc), as they both communicate only IS_4 to fighters, whereas the travel strategy travels signifi-



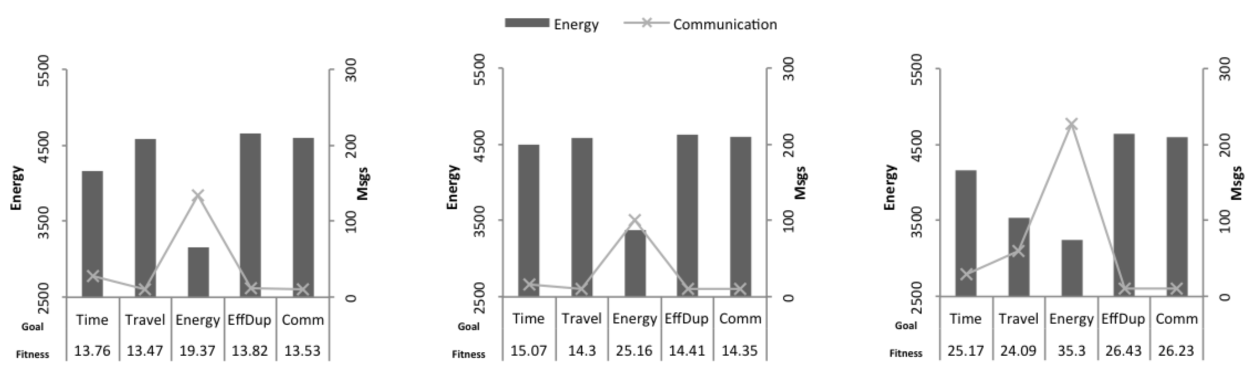
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.14: Time performance of communication strategies of various goals ($\alpha < \beta$)



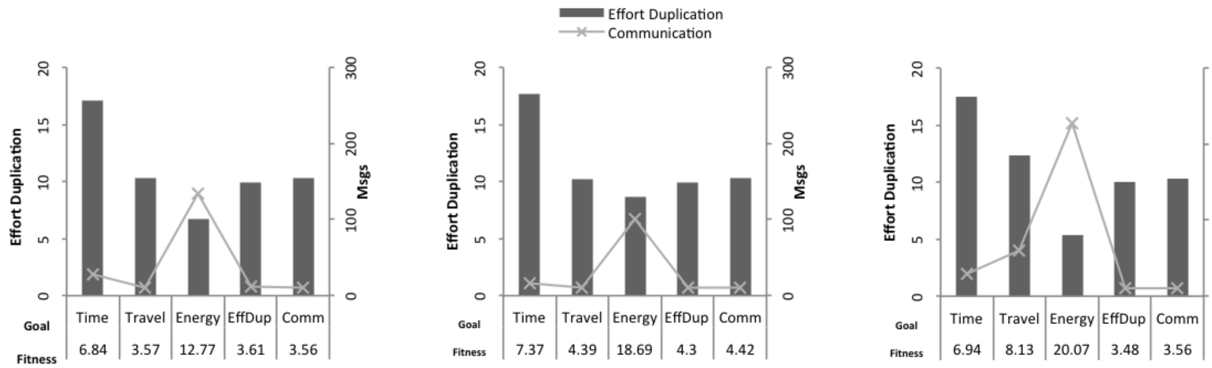
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.15: Travel performance of communication strategies of various goals ($\alpha < \beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.16: Energy performance of communication strategies of various goals ($\alpha < \beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 5.17: Effort duplication performance of communication strategies of various goals (α<β)

cantly shorter distance (p=0.005), Figure 5.15c, as a result of doubling the action cost that makes communicating IS_5 worthwhile.

When it comes to the energy performance (Figure 5.16), and aside from the admittedly energy strategy's performance, the time strategy wins the best performance, when (ac=cc), as it communicates more (p=0.003 comparing to the travel strategy). However, no significant difference in energy fitness is obtained (p=0.1). When action cost is high, travel strategy significantly outperforms the time strategy with respect to both energy and energy fitness, (p=0.004 and p=0.008, respectively). All strategies, except for energy, have roughly the same energy performance when the communication cost doubles as they all communicate minimally.

Figure 5.17 shows the effort duplication performance of the strategies. When action and communication costs are equal, time strategy has the worst performance because it communicates IS_4 to two fighters, does not communicate IS_8 to fighters, and communicates IS_8 to four carriers. All other strategies have the same performance as they all communicate minimally. When action cost is high, travel strategy has higher effort duplication than the other two cost settings, where its performance is similar to effort duplication and minimum communication strategies because of broadcasting IS_5.

5.2.3 Simulation Environment

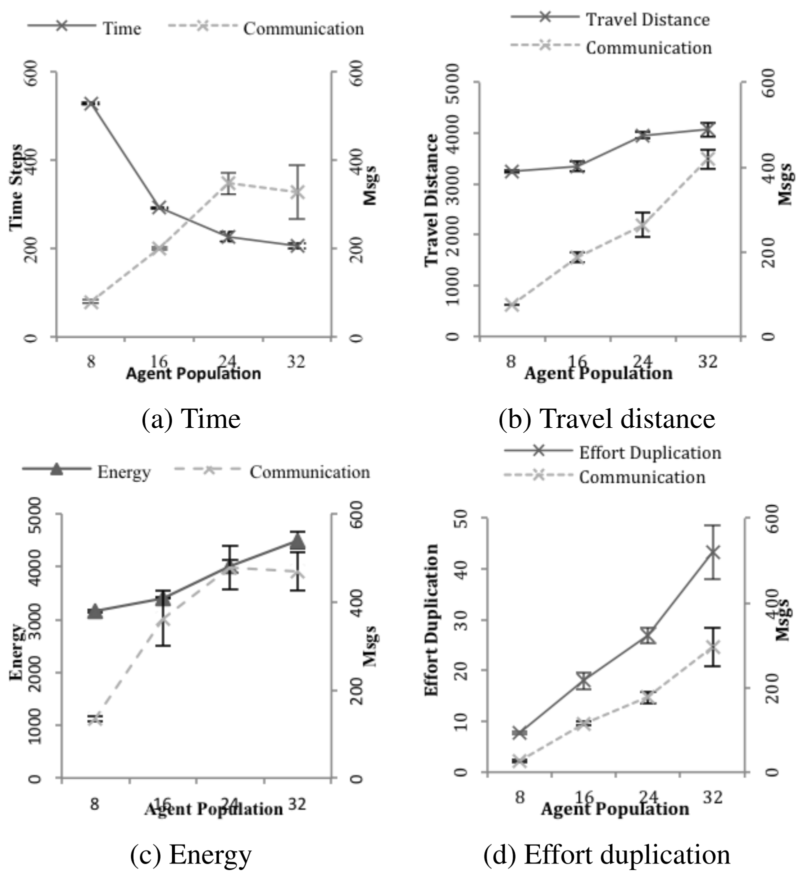
Next, we evaluate the algorithm in difficult scenarios, where the task complexity is greater than agent population, and simple scenarios, where the task complexity is less than agent population. In these experiments, we seek unbiased communication strategies where no preference is made between goal and communication cost. Therefore, we consider one specific case of fitness parameters where (α=β, ac=cc) to further investigate the impact of variation of simulation environment on the evolved strategies and goal performance.

In some simple scenarios, we notice that GA becomes less consistent with respect to the evolved strategies, by evolving neutral information instances in some runs, which results in higher standard error for the communication cost. In these cases, we report the strategy with best fitness.

5.2.3.1 Agent Population

Agent population in this domain is the total number of carriers and fighters, which is fixed at eight (five carriers and three fighters), in the basic setting adapted in all previous experiments. In this section, we show the performance of the proposed approach with larger populations.

Figure 5.18 illustrates the performance of evolved strategies, presented in Tables 5.6, 5.7, 5.8, and 5.9, for different populations, with respect to different performance goals. For time, increasing population to move the scenario from the case (TC>AP) to (TC=AP) enables finishing the task in significantly shorter time. However, extra agents, to move the scenario to simple case where (TC<AP), make no significant difference in the time consumed to finish the task (p=0.19, for 24 versus 32). Communication cost, however, significantly increases up to the first simple scenario, i.e., 24 agents, where GA was successfully able to evolve strategies with the same communication cost, although with higher standard error, which implies evolving neutral information instances (at 32). As shown in Table 5.6, the reason for the continuous increase in communication cost, only up to population of 24, is broadcasting empty rooms (IS_5) to all population 8, 16, 24, but restricting it to only 10 carriers in case of 32.



(a) Time (b) Travel distance

(c) Energy (d) Effort duplication

Figure 5.18: Various-goals performance with increasing agent population.

Multiple observations can be made about the changes that took place in the evolved time strategies in Table 5.6. We notice that the gold locations (IS_3) to carriers as well as the safe room (IS_8) to fighters become less important and hence are not communicated in strategies for larger populations. Instead, agents are allowed to communicate their goal room (IS_2), and are also encouraged to communicate the safe room (IS_8) to more carriers. When the number of carriers is small, we care more about guiding these agents to the gold locations, where they can go and collect it. However, when the number of carriers increases, the focus will be more on spreading these agents out to simultaneously explore as many rooms as possible, which is achieved by communication of goal rooms. This may cause that more carriers are exposed to wumpus rooms, which they would avoid for good, unless they are informed that a wumpus is killed, hence more recipients of IS_8. In a similar manner, when the number of fighters is small, and wumpus requests (IS_4) are broadcast, delay response to simultaneous requests are more frequent as fighters might be busy responding to a closer request. Therefore, a fighter is told when a request is completed (IS_8), such that it can fulfill another request. When the number of fighters is larger, although we still need to send requests to the same number of recipients (3-4 fighters), we no longer need to inform them about completed requests, as recipients of each request are probably different fighters.

In the travel distance goal, GA has successfully evolved strategies to maintain the same travel distance even with larger population in two cases; first when population increases from 8, where TC>AP, to 16, where TC=AP, and second from 24, where TC<AP, to 32, where TC<AP, by increasing communication. The total travel distance increases when the scenario moves from 16, where TC=AP to 24, where TC<AP.

Table 5.7 shows that communicating the wumpus location to carriers has less contribution to decreasing travel distance, and hence is dropped from the strategy, as population increases. The reason is that, in simple scenarios, the task is completed well before each carrier visits all rooms, hence sending the wumpus locations to carriers will not reduce the total travel distance, as carriers are less probable to reach idle state before the task is complete. Moreover, we observe that, although more fighters are added to the population, the travel strategy is consistent with the fighters' recipient strategy of IS_4, as no more fighters are needed to complete a request. Carriers are not allowed to communicate their goal room, except with large population, where it is worthwhile to do so to avoid carriers performing the same work.

Table 5.6: The evolved time communication strategies for larger populations.

Agent Population			Information Instances (IS_i)							
C Total	F		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C			2 (3 TS)		Bcast (EU)			P2P (9 TS)
		F				Bcast (5 TS)				P2P (14 TS)
10 16	6	C		P2P (0, 0)			Bcast (EU)			8 (EU)
		F				4 (6 TS)				
15 24	9	C		3 (0, 0)			Bcast (1, ϕ)			9 (5 TS)
		F				3 (EU)				
20 32	12	C		2 (ϕ, 0)			10 (13 TS)			14 (17 TS)
		F				3 (10 TS)				

Table 5.7: The evolved travel communication strategies for larger populations.

Agent Population			Information Instances (IS_i)							
C Total	F		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C				P2P (20 TS)	Bcast (EU)			P2P (11 TS)
		F				2 (1, 0)				
10 16	6	C					8 (1, 0)			3 (9 TS)
		F				P2P (EU)				
15 24	9	C					7 (10 TS)			3 (6 TS)
		F				P2P (EU)				
20 32	12	C		4 (1, 0)			14 (1, ϕ)			4 (7 TS)
		F				P2P (EU)				

In consistence with previous observations, communication costs obtained from multiple runs of GA, with energy as the performance goal, become even more variant as the agent population

Table 5.8: The evolved energy communication strategies for larger populations.

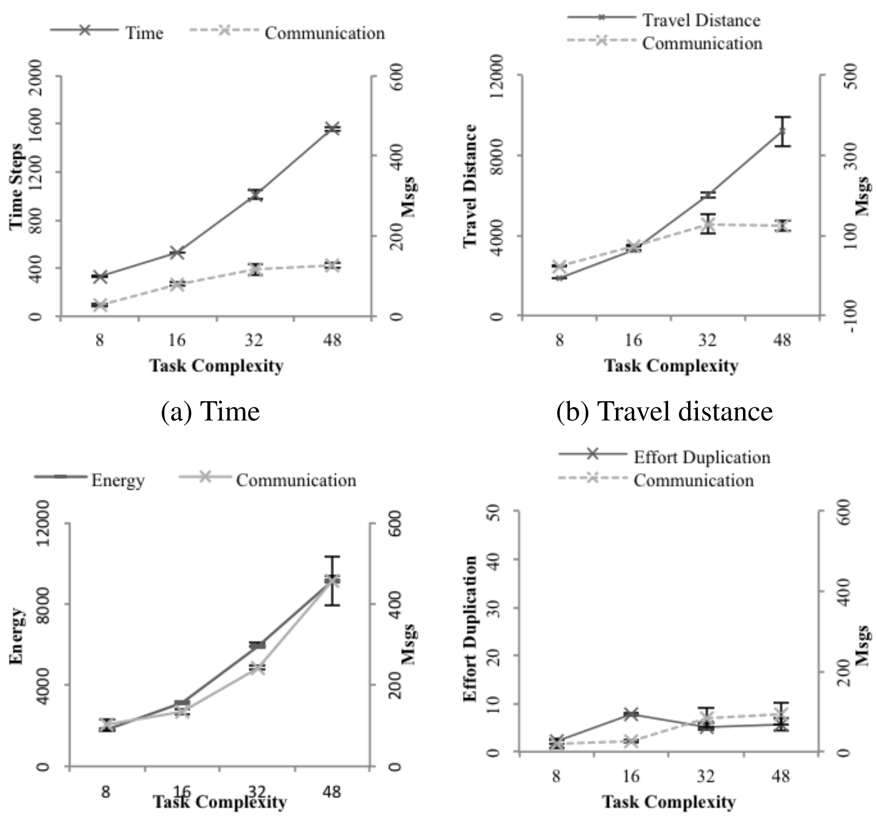
Agent Population C Total	F		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C		P2P (10 TS)		3 (7 TS)	Bcast (1, 0)			2 (1)
		F				P2P (EU)				
10 16	6	C	P2P (0, 0)	P2P (EU)	6 (12 TS)		Bcast (EU)			
		F				P2P (EU)	2 (13 TS)		P2P (4 TS)	
15 24	9	C		P2P (5 TS)	4 (2 TS)		13 $(1,\phi)$			12 (15 TS)
		F				2 (18 TS)				4 (7 TS)
20 32	12	C	P2P $(0, \phi)$	P2P $(0, \phi)$	14 (5 TS)		18 (EU)			
		F				2 (4 TS)		3 $(1, \phi)$		

Table 5.9: The evolved effort duplication communication strategies for larger populations.

Agent Population C Total	F		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C								P2P (19 TS)
		F				2 (7 TS)				P2P (2 TS)
10 16	6	C		5 (0, 0)	6 (1, 1)					6 (EU)
		F				2 (8 TS)				3 (11 TS)
15 24	9	C		2 (2 TS)	3 (19 TS)					
		F				P2P (EU)				
20 32	12	C		6 $(\phi, 0)$	13 (8 TS)					
		F				2 $(1, \phi)$				

increases. Energy strategies are provided in Table 5.8. Similar to travel strategies, communicating wumpus locations to carriers is dropped as population grows, and similar to effort duplication, gold location is allowed in larger population, in effort to minimizing time and hence the amount of energy consumed.

The effort duplication strategies go through two main changes as the population increases, in Table 5.9. The first change happens at population size of 16, as carriers' goal room (IS_2) and gold locations (IS_3) are added to the strategy. As explained before, duplication of effort can take three forms. First, when two or more carriers explore a room simultaneously. Second, when two or more carriers target the same piece of gold. Third, and last, is when two fighters or more target the same wumpus. In all sizes of population, duplication of fighters' efforts can be prevented, or minimized, by avoiding broadcasting wumpus locations to fighters. For duplication of carriers' efforts, however, both gold collection duplication as well as room exploration duplication occur in small population. As the population increases, with fixed task complexity, majority of effort duplication occurs as room exploration, rather than gold collection. Therefore, agents are allowed to communicate their goal rooms. In the original small population size, though, this information instance is not communicated unless goal outweighs communication, due to its high cost. In larger population, carriers are actually encouraged to communicate gold locations with many neighbors, although this might increase gold collection duplication slightly, since the world contains only a few pieces of gold, it accelerates task completion, and hence minimizes duplication of room exploration. In the case when agent population and task complexity are equal, fighters are still allowed to communicate safe rooms with many carriers, as it helps prevent carriers from overcrowding in non-wumpus rooms. The impact of this information instance reduces as population grows, because wumpuses are killed instantly due to the availability of more fighters; hence fewer carriers are exposed to wumpuses.



(a) Time (b) Travel distance

(c) Energy (d) Effort duplication

Figure 5.19: Various-goals performance with increasing task complexity

5.2.3.2 Task Complexity

This feature refers to how complex the task is to agents, represented by the amount of gold and number of wumpuses present in the world. Figure 5.19 depicts the system's performance with respect to various goals, as the task complexity increases, while the evolved strategies are presented in Tables 5.10, 5.11, 5.12, and 5.13. The basic task complexity, used in all previous experiments, is the second case, i.e., 16.

Unlike the previous feature, the time needed to complete the task experiences a steep increase as the task gets more complex, while the communication curve has gentle gradient, as a result of applying evolved strategies. The same trend applies to the travel distance goal, although it experiences a greater error rate in the most complex scenario.

Aside from communicating IS_8 to fighters in the strategy for the basic task complexity, time

Table 5.10: The evolved time communication strategies for more complex tasks.

Task Complexity			Information Instances (IS_i)							
Gold Total	Wumpus		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C			2 (17 TS)		P2P (20 TS)			3 (2 TS)
		F				2 (1, ϕ)				
10 16	6	C			2 (3 TS)		Bcast (EU)			P2P (9 TS)
		F				Bcast (5 TS)				P2P (14 TS)
20 32	12	C			2 (1, 0)		P2P (10 TS)			P2P (15 TS)
		F				Bcast (2 TS)				
30 48	18	C			3 (14 TS)		P2P (1, ϕ)			P2P (2 TS)
		F				2 (EU)				

Table 5.11: The evolved travel communication strategies for more complex tasks.

Task Complexity			Information Instances (IS_i)							
Gold Total	Wumpus		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C			2 (17 TS)		P2P (20 TS)			P2P (12 TS)
		F				P2P (EU)				
10 16	6	C				P2P (20 TS)	Bcast (EU)			P2P (11 TS)
		F				2 (1, 0)				
20 32	12	C				P2P (8 TS)	Bcast (1, ϕ)			P2P (13 TS)
		F				P2P (EU)				
30 48	18	C				P2P (EU)	3 (1, 0)			P2P (1, ϕ)
		F				P2P (EU)				

Table 5.12: The evolved energy communication strategies for more complex tasks.

Task Complexity			Information Instances (IS_i)							
Gold Total	Wumpus		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C		P2P (15 TS)	4 (5 TS)		3 (1, ϕ)			P2P (17 TS)
		F				P2P (EU)				
10 16	6	C		P2P (10 TS)		3 (7 TS)	Bcast (1, 0)			2 (1)
		F				P2P (EU)				
20 32	12	C		P2P (4 TS)	2 (12 TS)	Bcast (9 TS)	Bcast (4 TS)			P2P (16 TS)
		F				P2P (EU)				
30 48	18	C		P2P (4 TS)	2 (13 TS)	2 (13 TS)	Bcast (1, 0)			P2P (15 TS)
		F				P2P (EU)		2 (1, ϕ)		

Table 5.13: The evolved effort duplication communication strategies for more complex tasks.

Task Complexity			Information Instances (IS_i)							
Gold Total	Wumpus		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8
5 8	3	C		P2P (0, 0)						
		F				P2P (16 TS)				
10 16	6	C								P2P (19 TS)
		F				2 (7 TS)				P2P (2 TS)
20 32	12	C		2 (0, 0)						P2P (17 TS)
		F				P2P (EU)				
30 48	18	C		P2P (0, 0)						P2P (15 TS)
		F				P2P (EU)				

strategies are consistent, with respect to the communicated information instances, for different cases of task complexity (Table 5.10). However, we notice a difference in the recipient strategy of IS_5. We have seen that, in the basic task complexity, IS_5 is always broadcast to carriers whenever communication is allowed, in all parameter settings. When the task becomes less complex 8 or more complex 32, 48, however, IS_5 is communicated only to the closest carrier. The reason is that, on one hand, the world in the simple scenario contains only five pieces of gold, hence if carriers inform each other about the locations of these gold, they will be able to complete the task well before they consider exploring an empty room. On the other hand, difficult scenarios incorporate large amount of gold, i.e., 20 and 30, in the world, which makes the possibility of having an empty room in the early stage of task execution, i.e. when carriers explore rooms, very low as the world includes only 12 rooms, and hence makes broadcasting it not significant to minimizing time. Therefore, the focus, when minimizing time is to share gold locations, rather than empty rooms.

Similarly, in the simple scenario of the travel distance goal, where only a small amount of gold is present in the environment, carriers are allowed to communicate the gold locations to the closest two carriers. It seems that, in such cases, completing the task as quick as possible is the best approach to minimize the distance traveled by all agents, as in such situations the task might be completed even before all wumpuses are killed.

The logic is different, however, in more complex scenarios. Communicating gold locations is avoided, as the focus is more on helping each carrier do the work in nearby rooms by communicating safe rooms as P2P, and avoiding empty and wumpus rooms by broadcasting empty rooms (IS_5), and communicating wumpus locations (IS_4).

Similar to time and travel distance, the amount of energy consumed experiences a steep increase as the task complexity increases. However, unlike the two performance goals, the amount of messages exchanged increases significantly, as well, with more complex scenarios. We believe that the same, or close, performance could be achieved with less communication, especially in the two most complex scenarios.

Figure 5.19d illustrates that the amount of effort duplication increases when the scenario moves from simple 8, where TC=AP, to difficult 16, where TC>AP. In the simple scenario, amounts of gold and number of carriers are equal and numbers of wumpuses and fighters are equal. Therefore, we can avoid fighters' effort duplication by sending wumpus locations (IS_4) to one fighter, and similarly, we can avoid carriers' efforts duplication by allowing communication of goal rooms (IS_2), as shown in Table 5.13.

When task complexity doubles with the same agent population, agents take longer time to complete the task, and hence change goal rooms more often than the previous case. GA disallows communication of goal rooms as it is too expensive, for the current weight settings, which results in increase in effort duplication. Rather, as the world contains more wumpuses, and hence more carriers are likely to observe them and avoid their rooms, GA evolves communication of safe rooms (IS_8) to only the closest carrier. In addition, to prevent recipients of a wumpus location to target the same wumpus, IS_8 is communicated to fighters as well.

As the task complexity doubles once more and twice, 32, 48, communication of goal rooms seems to be worthwhile again. The reason is that as the amount of gold and wumpuses increases, time to complete increases as well, and hence carriers are more likely to duplicate each other's work. Safe rooms are still allowed to be communicated to the closest carrier, for the same aforementioned reason.

5.3 Summary

Our findings indicate that GA has successfully evolved strategies for communicating information that contribute to improving the goal performance of the system according to the desired parameters. Applying the weighted sum method in the fitness function to make trade-off between the goal performance and communication cost has played an important role in the success of this approach, as results produced form communication-inclusive goal (energy, in this case) were less promising due to over communication.

We have shown that information instances can be classified, with respect to each goal performance, into favorable, unfavorable, or neutral information instance. We have also shown that the the class that an information instance belongs to or their importance to a goal performance may change in response to changes in the simulation environment. The favorability of communicating an information instance is said to be reduced when it is not allowed to be communicated or communicated to fewer recipients, compared to the original case. The favorability of communication is considered increased when the opposite holds true.

In the case of time performance, communicating the information instances IS_2, IS_3, IS_5, and IS_8 to carriers and IS_4 to fighters are considered favorable in the original case. Yet, we observed that with larger agent populations the importance of communicating IS_2 and IS_8 to carriers increased, while the importance of IS_3 to carriers and IS_4 to fighters decreased as they are communicated to fewer recipients or not communicated at all. Additionally, in more complex scenario, we observed decrease in the importance of communicating IS_5 to carriers.

When the goal is to reduce the total travel distance, the information instances IS_2, IS_4, IS_5, and IS_8 to carriers and IS_4 to fighters are considered favorable. As the agent population increases, the favorable impact of communicating IS_4 and IS_5 to carriers decreased, while the impact of communicating IS_2 and IS_8 to carriers increased. With more complex scenarios, the influence of communicating IS_5 to carriers increased.

In the case of effort duplication, our results have shown that favorable information instances include IS_2, IS_5, and IS_8 to carriers and IS_4 and IS_8 to fighters. With larger populations, the favorable impact of communicating IS_2 and IS_3 to carriers increased, while the favorable impact of communicating IS_8 to both carriers and fighters decreased. In more complex scenarios, we observed decrease in the impact of communicating IS_8 to fighters and increase in favorability of communicating IS_2 to carriers.

The evolved strategies avoided invalid, unfavorable, and neutral information. However, assigning a relatively low weight to communication cost or evolving a strategy for simple scenarios may sometimes result in higher standard error for communication, as GA may evolve a communication strategy that allows communicating neutral information. Furthermore, we observed that increasing weight for the performance goal may not necessarily result in better performance, as it depends on the cost of communication needed to further improve the performance, and whether it results in better fitness, given the assigned weights.

CHAPTER 6

Case Study: Collective Construction

6.1 Scenario

WE HAVE designed, developed, and implemented a Collective Construction domain using Repast Simphony (North et al., 2005). The simulation environment is a grid of 140x140 cells, which includes two zones; the Main Search Zone (*MSZ*), which represents the whole world, and the Construction Zone (*CZ*), which is a sub-zone of the former, located in the middle and occupies a space of 50x50 cells. The search zone represents the area where the building blocks, which come in six different colors, are randomly distributed at the beginning of each simulation. Inside the construction zone is a H-shaped map, where the color of each site corresponds to the block that needs be attached. The construction zone, including sites that belong to the structure, namely, the Structure Sites (*SS*), are covered randomly by debris. Around the construction zone exists six small zones, which are sub-zones of *CZ*, and called the Supply Zones (*SZ*). Each supply zone has a maximum capacity of three blocks. The color of each supply zone corresponds to the blocks that can be placed.

The main search zone contains five collectors and two hundreds blocks, divided equally into six different colors. The construction zone contains three builders, five bulldozers, and amount of a hundred debris, all of which are distributed randomly, in the corresponding zone, at the beginning of each simulation experiment. Each agent type is assigned a task out of three main tasks required for the construction of a structure, including preparation, collection, and building. First, bulldozers are responsible for preparing the construction site by clearing it from debris. They do so by performing random walk to find and remove debris. Each bulldozer is able to carry a maximum amount of 20 debris, in which case it has to dispose of the debris at the borders of the search zone. Second, collectors are in charge of searching for, collecting, and delivering of the building materials to the supply zones. Third, builders build the structure by consulting the shape map, visiting a supply zone, collecting a block, and placing the block into the designated structure site. The construction process of the structure is depicted in Figure 6.1. Gray area is the construction site and dark x's are debris. Structure sites and supply zones are highlighted with the color of blocks to be occupied with; Builders are depicted as black circles, collectors are red circles, and bulldozers are yellow circles. The information instances that agents can communicate in this domain are

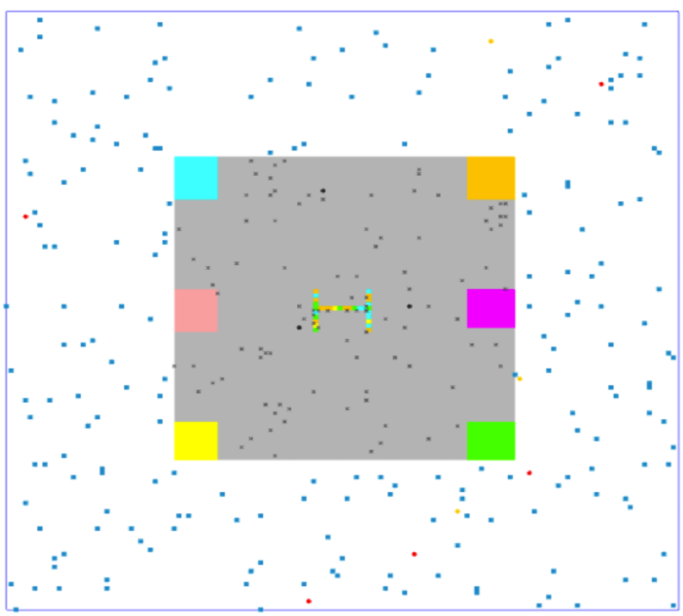


Figure 6.1: Snapshot of the Collective Construction of an H-shaped structure.

listed in Table 6.1. Unlike the Wumpus World domain, all information instances of this domain are considered multi-value, where an agent may have a list of all values that belong to an information instance. Further, information instances in this domain are categorized into two groups, according to their durability.

1. **Long-term**: Information instances, whose values once obtained are irreversible. Examples include structure updates (IS_2), sufficient supply (IS_4), and clear structure sites (IS_{12}).

2. **Short-term**: Information instances, whose values once obtained, are valid temporarily. Examples include empty supply zones (IS_6) and full supply zones (IS_{11}), as supply zones are visited regularly by both builders and collectors who can alter their contents. For this purpose, we designed builders and collectors to forget values of these information instances that are one hundred time-steps old.

Table 6.1: Information instances of the Collective Construction domain and their values.

Identifier	**Information Instance**	**Possible Values**	**Producer**
IS_1	Target	(x,y),where: $0 < x, y < 140$	Builder
IS_2	Structure Update	(x,y),where: $x, y \in SS$	Builder
IS_3	Insufficient Supply	(sz, n), where: sz=[1,6] n= *needed-present*	Builder
IS_4	Sufficient Supply	[1, 6]	Builder
IS_5	Decrease in a Zone's Supply	(sz, n), where: sz=[1,6] n: sz size	Builder
IS_6	Empty Supply Zone	[1,6]	Builder
IS_7	Debris Location	(x,y) $x, y \in SS$	Builder
IS_8	Block Location	(x,y) $x, y \in MSZ$	Collector
IS_9	Collected Block Location	(x,y) $x, y \in MSZ$	Collector
IS_{10}	Increase in a Zone's Supply	(sz, n), where: sz=[1,6] n: sz size	Collector
IS_{11}	Full Supply Zone	[1,6]	Collector
IS_{12}	Clear Structure Site	(x,y) $x, y \in SS$	Bulldozer

At each time-step, every agent performs (observe, communicate, act) task cycle. In observe, agents are able to make observations. For example, bulldozers can observe debris and clear structure sites (IS_{12}), collectors can observe blocks (IS_8), and builders can observe structure updates (IS_2), debris (IS_7), and empty supply zones (IS_6). In communicate, agents consult their communication strategy to decide whether they should communicate at the current time-step. In the act step, all agents are able to move UP, DOWN, RIGHT, and LEFT as well as PICK UP, and DROP OFF objects, whether blocks or debris.

Additionally, all agents maintain inner states. Builders can be in one of four states: *Find*, *obtain*, *build*, and *dispose*. All builders begin in the *Find* state, where one decides which supply zone to visit next to find a block. Once a builder encounters the desired block, it transits to the *obtain* state. A builder remains in this state until it picks up a block, when a transition to the *build* state is triggered. If a builder succeeds in attaching a block to the designated structure site, it switches to the *Find* state. However, if a builder finds a debris or a block in the structure site, and is unable to find another proper site, it is considered fail and the builder transits to the *dispose* state. In this state, a builder transports the block back to the supply zone, and then transits to the *Find* state. Builders update their shape map with every visit to the structure, or if they receive information about structure updates (IS_2) or debris location (IS_7).

Collectors can be in one of four states: *wander*, *obtain*, *deposit*, and *dispose*. A collector starts and remains in the *wander* state, where it performs random walk, until it observes a block. This triggers transit to the *obtain* state, as a collector targets and collects the block. Next, the collector moves to the *deposit* state, in which a collector travels to the designated supply zone and places the block. If a collector is able to place a block in the *SZ* successfully, it switches to the *wander* state. However, if the collector fails at placing the block due to the *SZ* being full, it transits to the *dispose* state, where it goes to the search zone and drops off the block at a random location.

Bulldozers can be in one of three states: *wander*, *clear*, and *dispose*. Similar to collectors, bulldozers begin in the *wander* state, as they walk randomly, until debris is observed. If a bulldozer senses debris, it transits to the *clear* state, where it targets the debris and removes it. Each bulldozer can carry a maximum amount of 20 debris. Should a bulldozer reach its full capability, it transits to the *dispose* state, as it moves to the borders of the search zone and gets rid of the debris that it carries. Once this is complete, or if the bulldozer is still able to clear more debris, transition to the *wander* state is triggered. Figures 6.2, 6.3, and 6.4 show the Finite State Machines (FSM) for builders, collectors, and bulldozers, respectively.

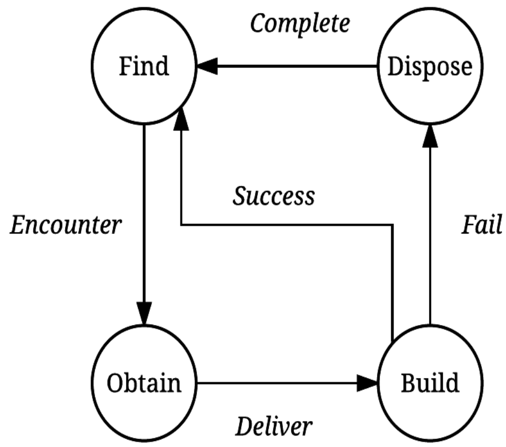


Figure 6.2: Builder FSM.

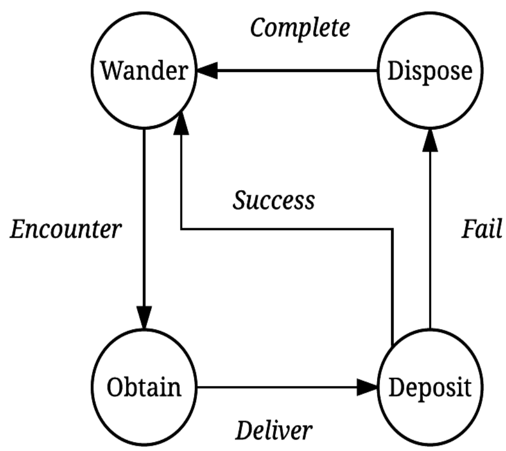


Figure 6.3: Collector FSM.

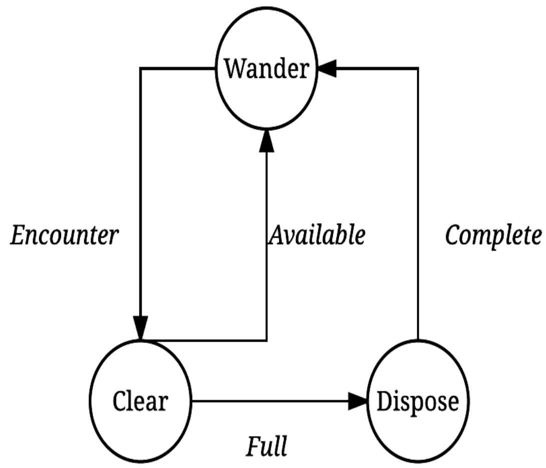


Figure 6.4: Bulldozer FSM.

6.2 Experimental Results

6.2.1 Fitness Parameters

The experimental results for each performance goal with varied values of the fitness parameters are presented in Figures 6.5, 6.6, 6.7, and 6.8. As shown, each bar represents the goal performance for one fitness parameter setting, which has been hierarchically labeled by the goal and communication weights and further by the action and communication costs. The line chart for each cost setting represents the communication cost trend as the goal's weight increases. The performance results reported are the average of three evolution runs. For each result, the error bar shows the standard error, computed as STDEV/SQRT (n), across n GA runs, where n=3.

Further, the evolved communication strategies are shown in Tables 6.2, 6.3, 6.4, and 6.5. Each row shows the strategy evolved for one parameter setting, and each column represents an information instance. Dark shaded cells represent invalid communication, or communication that only increases cost with no contribution to performance improvement. Such communication includes sending structure's updates to bulldozers or collectors, and debris locations to collectors. Light shaded cells convey that no communication is allowed for the corresponding information instance in the corresponding parameter setting (strategy). For clear cells, the upper line represents the recipient strategy, i.e., P2P, Bcast, or subset [2-max], and the lower line represents the timing strategy, which can take either every update (EU), in a state {*find*, *obtain*, *build*, *dispose*} for builders (B), {*wander*, *clear*, *dispose*} for bulldozers (Bz), and {*wander*, *obtain*, *deposit*, *dispose*} for collectors (C), or every time interval ([2-20] TS), as explained previously.

6.2.1.1 Time

We present the time performance for the evolved strategies with different values for the fitness parameters in Figure 6.5. We notice that communication is very responsive to its assigned weight as lower weight always results in higher communication and higher weight results in lower communication, yet this does not always hold true for time.

Theoretically, the minimum amount of communication, and hence worst performance, is evolved for the parameter setting ($\alpha<\beta$, ac<cc), which is the case when communication is assigned high weight and high cost. Figure 6.5 shows that this is attained for this parameter setting, as well as the two corresponding cases when communication cost is not high ($\alpha<\beta$, ac=cc) and ($\alpha<\beta$, ac>cc), with (p=0.8). As shown in Table 6.2, GA evolved close strategies for these parameter settings, where sufficient supply (IS_4) is communicated to collectors and debris location (IS_7) is communicated to bulldozers. We notice that when communication cost is high ($\alpha<\beta$, ac<cc), IS_4 is communicated as P2P, i.e., to only the closest collector, while in the other two cases, IS_4 is communicated to a subset of the collectors. However, in the case of high communication cost, IS_4 is communicated more often. When IS_4 is communicated to collectors, builders send messages when they have sufficient supply of blocks of one color. As explained previously, although building blocks are available in six colors, only three colors are picked randomly at the beginning of each simulation run to be used to assign a color to each structure site at random. As the shape map is only accessible to builders, collectors may spend their time collecting blocks that are not needed. Other cases include when all structure sites belonging to a color have already been occupied, or

when the available blocks in a supply zone are equal to or more than what is needed. Communicating this information allows collectors to use their time efficiently by focusing on collecting blocks of other colors that are still needed, hence providing blocks faster to the supply zones. This means that builders experience no delay in building the structure as they always have available supply of needed blocks. Debris location (IS_7) allows builders to send structure sites to bulldozers where debris has been observed. Due to the large amount of debris present in the environment, bulldozers may take long time to find the debris in structure sites and clear them, and hence the task of building the structure may consequently take longer time as builders have to wait until debris is cleared to be able to attach blocks in the corresponding sites. Communicating IS_7 helps avoid this scenario as builders notify bulldozers about debris locations in the structure.

The next time performance is achieved in the parameter setting when communication cost is high with equal weights (α=β, ac<cc). In this case, in addition to communicating IS_4 to more collectors and IS_7 to more bulldozers, builders are allowed to communicate structure updates (IS_2) with each others; and bulldozers are allowed to communicate clear structure sites (IS_{12}) to builders. T test suggests that communicating IS_2 and IS_{12} has improved the time performance significantly (p=0.002), which implies that the two information instances are favorable to time performance. When the goal's weight is increased to switch to the case (α>β, ac<cc), agents are allowed to communicate more due to the lower communication's weight (Figure 6.5), yet the improvement in the system performance is not significant (p=0.05), compared to the case (α=β, ac<cc). In addition to the information instances communicated in the previous case (α=β, ac<cc), namely, IS_2, IS_4, IS_7, and IS_{12}, builders are allowed to communicate insufficient supply (IS_3) to the closest collector. It is believed that this information instance may be valuable to time in some cases, such as when a block is needed to be attached to a structure site, but the corresponding supply zone is empty. In this case, sending insufficient supply (IS_3) for this block to a collector can lead to quick completion of the task.

It can be observed in Figure 6.5 that corresponding strategies at the two weight settings (α=β) and (α>β) when communication cost is not high (ac=cc) and (ac>cc) have close time performance (p=0.26 and p=0.38, respectively), since increasing action cost has no effect on time fitness. By comparing the two communication strategies when (α=β), in Table 6.2, we notice that both strategies are very similar as both broadcast IS_2, IS_{10}, and IS_{12} to builders, communicate IS_4 to three collectors, and communicate IS_7 to two bulldozers. Yet, while the strategy (α=β, ac=cc) communicates IS_{10} to collectors, the strategy (α=β, ac>cc) allows communicating IS_3 to them. It appears that the two information instances have close impact on time as the former keeps collectors informed about the status of the supply zones, and the latter gives them updates about supply needed to complete the task.

Further, we notice that the system performs closely when the time weight increases to switch from (α=β) to (α>β) in both cost settings (p=0.24 and p=0.32, respectively). The ANOVA results allow us to state with 95% confidence that all four communication strategies, namely, (α=β, ac=cc), (α>β, ac=cc), (α=β, ac>cc), and (α>β, ac>cc), are equivalent (p=0.66). By comparing the four communication strategies in Table 6.2, we observe that the two strategies with high time's weight (α>β) communicate all information instances that are communicated by the two strategies with lower time's weight (α=β) with close recipient and timing strategies. However, the former strategies further communicate other information instances, such as IS_5, IS_6, and IS_{11}. As four strategies perform closely with respect to time, these information instances can be considered neutral to time performance as they were allowed to be communicated due to the low communication's cost and weight and fail to improve performance. The four strategies achieve the fastest time performance, and significantly outperform strategies with slowest performance (α<β) with (p<0.00001) and strategies with the next longest time performance (α=β,ac<cc) and (α>β, ac<cc) with (p<0.00001).

As shown in Table 6.2, the information instances IS_4, IS_7 are communicated in all strategies, yet more often or to more recipients in some cases. For example, IS_7 is communicated as P2P when (α<β), but to more bulldozers when (α=β) and (α>β). Therefore, we conclude that these are the most influential information instances to time performance in this domain. The second important information instance to time seems to be structure updates (IS_2) and clear structure spots (IS_{12}) as they are communicated in six strategies, when communication's weight is equal to/less than the goal's weight. When builders share structure updates with each other, they will waste no time trying to place a block in a spot already occupied. It appears that the next influential information is the insufficient supply (IS_3) and zone supply increase (IS_{10}), as both are communicated in four strategies. Three information instances, namely, builders' targets (IS_1), block location (IS_8) and collected block location (IS_9) were not communicated at all as they either have no effect on time, i.e., neutral, or are too expensive.

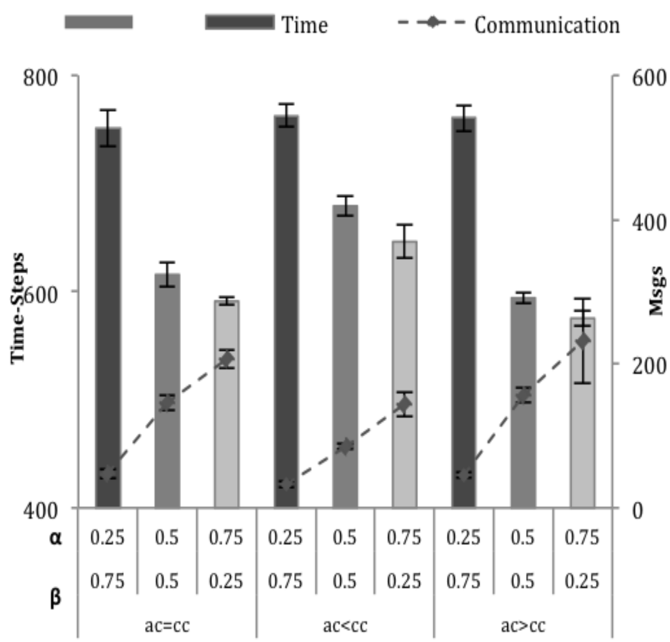


Figure 6.5: Performance of time strategies.

Table 6.2: The evolved time communication strategies.

Fitness Parameters			Information Instances (IS_i)											
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
ac=cc	$\alpha<\beta$	B												
		Bz							P2P (18TS)					
		C				2 (18TS)								
	$\alpha=\beta$	B		Bcast (14TS)								Bcast (Wander)		Bcast (10TS)
		Bz							2 (14TS)					
		C				3 (7TS)						2 (Wander)		
	$\alpha>\beta$	B		Bcast (13TS)			P2P (Build)	P2P (18TS)				Bcast (4TS)	P2P (13TS)	Bcast (18TS)
		Bz							4 (3TS)					
		C			P2P (15TS)	Bcast (15TS)	P2P (9TS)							
ac<cc	$\alpha<\beta$	B												
		Bz							P2P (Build)					
		C				P2P (6TS)								
	$\alpha=\beta$	B		Bcast (7TS)										2 (12TS)
		Bz							2 (Build)					
		C				3 (17TS)								
	$\alpha>\beta$	B		Bcast (9TS)										P2P (15TS)
		Bz							P2P (5TS)					
		C			P2P (Find)	2 (14TS)								
ac>cc	$\alpha<\beta$	B												
		Bz							P2P (20TS)					
		C				3 (20TS)								
	$\alpha=\beta$	B		Bcast (2TS)								Bcast (15TS)		Bcast (7TS)
		Bz							2 (20TS)					
		C			P2P (EU)	3 (EU)								
	$\alpha>\beta$	B		Bcast (6TS)			P2P (20TS)	P2P (18TS)	P2P (18TS)			Bcast (20TS)		2 (10TS)
		Bz							2 (5TS)					
		C			2 (20TS)	3 (8TS)	P2P (Build)	3 (Build)						

6.2.1.2 Travel Distance

The performance of the travel strategies is presented in Figure 6.6. A major performance difference in this goal compared with time is that increasing action cost does affect fitness. Therefore, we observe asymmetrical performance for the two cost settings (ac=cc) and (ac>cc). When comparing the performance in cases where communication's weight is low (α<β) across different cost settings, we notice that the maximum distance traveled is when communication has higher cost than action (ac<cc), as agents communicate minimally, where only sufficient supply (IS_4) and debris location (IS_7) are communicated, followed by the case when communication and action have equal costs (ac=cc), as clear structure site (IS_{12}) is further communicated to builders, and lastly when communication has lower cost than action (ac>cc), where both structure updates (IS_2) and empty supply zone (IS_6) are communicated to builders as well as communicating IS_4, IS_7, and IS_{12}. ANOVA test suggests that the performance difference between the three strategies is significant (p=0.007), which confirms the importance of communicating IS_2, IS_6, and IS_{12} to builders on travel performance.

As shown in Figure 6.6, three strategies of different parameter settings, namely (α>β, ac<cc), (α=β, ac=cc), and (α=β, ac<cc) have close travel performance to the latter strategy (α<β, ac>cc), confirmed by ANOVA test (p=0.69). The former two strategies along with (α<β, ac>cc) support the argument that inverse variations between the weight and cost of goal and communication can even out their importance, which results in similar strategies and performance.

The evolved strategies for these parameter settings (Table 6.3) indicate that they all allow communicating five information instances, of which four are mutually communicated, namely, IS_2, IS_4, IS_7, and IS_{12}. Each strategy communicates a different fifth information instance, which eventually results in close travel performance. Minimizing the total distance traveled by agents can be achieved by minimizing the distance traveled by builders, bulldozers, collectors, or all of them. In the four equivalent communication strategies, information instances are communicated to all agent types to improve their travel performance, yet with one unique component. The strategy (α=β, ac=cc) communicates builders target (IS_1) to builders to avoid having two builders that target the same structure site or supply zone, the strategy (α=β, ac<cc) communicates empty supply zone (IS_6) to collectors to encourage them to deposit supply if needed, while (α<β, ac>cc) communicates IS_6 to builders to help them avoid empty supply zones, and (α>β, ac<cc) communicates debris locations IS_7 to builders to help them avoid targeting structure sites that contain debris for attaching a block.

As we increase the travel's weight in the two equivalent strategies (α=β, ac=cc) and (α<β, ac>cc) to switch to the two strategies (α>β, ac=cc) and (α=β, ac>cc), the distance traveled decreases significantly (p=0.0008 and p=0.014), respectively, and we encounter another two equivalent strategies that support the aforementioned argument of the effect of inverse variations, confirmed by t test with p=0.42. Despite the difference in values of the fitness parameters in the two strategies, travel distance remains the more valuable goal in both cases. In the former strategy, travel distance has greater weight that communication, but action and communication costs are equal, whilst the opposite holds true in the latter strategy.

Table 6.3 shows that both communication strategies (α>β, ac=cc) and (α=β, ac>cc) communicate IS_2, IS_4, IS_7, IS_{10}, and IS_{12}. Yet, while the former communicates decrease in zone's supply (IS_5) to builders and collected block's location (IS_9) to collectors, the latter communicates empty supply zone (IS_6) to builders and block's location (IS_8) to collectors. We also notice, in Figure 6.6, that strategy (α=β, ac>cc) is able to achieve close travel performance, although more variant, to the strategy (α>β, ac=cc), but with less communication. The reason is due to the equal weights assigned for travel distance and communication in the strategy (α=β, ac>cc), as oppose to the low communication weight in the strategy (α>β, ac=cc). This made GA yield that, unlike (α>β, ac=cc), keeping all builders updated about the status of the supply zones, i.e., broadcasting IS_5, is not necessary in the strategy (α=β, ac>cc) to achieve the same travel performance. Rather, GA evolved informing only the closest builder when a supply zone is empty (IS_6) to help the builder make better zone decisions.

The strategy that wins the best performance is the case when the travel distance is assigned high weight and high cost (α>β, ac>cc). T test suggests that travel distance has decreased significantly (p=0.03), compared to the previous strategy (α=β, ac>cc). As shown in Table 6.3, this strategy broadcasts most information instances to builders as well as communicating IS_5, IS_6, and IS_9 to collectors. Based on the aforesaid observations and analysis, it is believed that the most influential information instances are IS_4 and IS_7 as they are communicated in all parameter settings. The next most influential information instance appears to be IS_{12} as it is dropped from only one strategy, followed by IS_2 as it is dropped from two strategies. Communicating IS_{10} and IS_6 to builders seems to be the next important information instance as they are communicated in three strategies, and then IS_1 to builders, and IS_6 and IS_9 to collectors. The information instances with the least impact on travel performance appear to be IS_5, IS_8, IS_{11} to collectors as they are communicated in only one case. Unlike time, IS_3 is not allowed to be communicated at all.

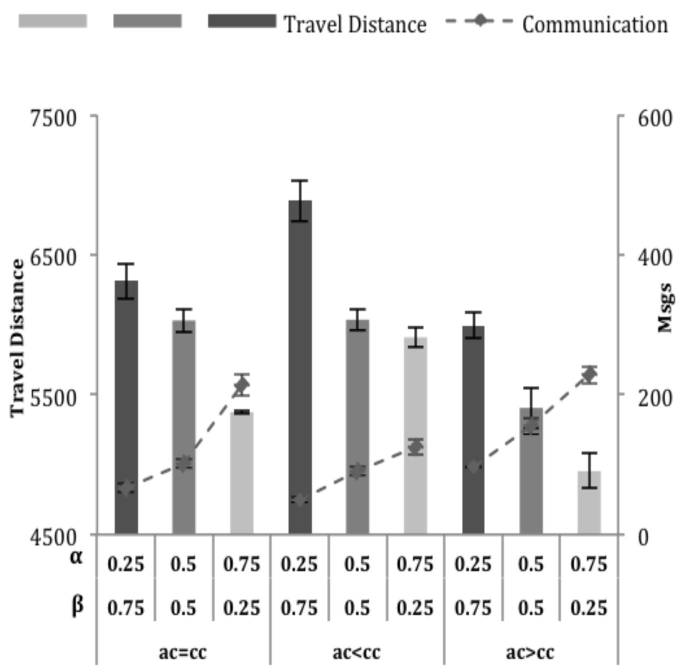


Figure 6.6: Performance of travel strategies.

Table 6.3: The evolved travel communication strategies.

Fitness Parameters			Information Instances (IS_i)											
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
ac=cc	$\alpha<\beta$	B												2 (17TS)
		Bz							P2P (4TS)					
		C				Bcast (EU)								
	$\alpha=\beta$	B	Bcast (Find)	Bcast (2TS)										Bcast (4TS)
		Bz							P2P (4TS)					
		C				Bcast (3TS)								
	$\alpha>\beta$	B		Bcast (7TS)			Bcast (3TS)					Bcast (9TS)		Bcast (EU)
		Bz							4 (18TS)					
		C				Bcast (2TS)					3 (14TS)			
ac<cc	$\alpha<\beta$	B												
		Bz							P2P (18TS)					
		C				Bcast (2TS)								
	$\alpha=\beta$	B		Bcast (8TS)										Bcast (Wander)
		Bz							P2P (17TS)					
		C				Bcast (Find)		P2P (6TS)						
	$\alpha>\beta$	B		Bcast (13TS)					P2P (Obtain)					P2P (9TS)
		Bz							2 (8TS)					
		C				Bcast (6TS)								
ac>cc	$\alpha<\beta$	B		Bcast (11TS)				P2P (2TS)						P2P (12TS)
		Bz							P2P (2TS)					
		C				Bcast (19TS)								
	$\alpha=\beta$	B		Bcast (5TS)				P2P (13TS)				Bcast (5TS)		2 (19TS)
		Bz							2 (15TS)					
		C				Bcast (EU)				2 (Wander)				
	$\alpha>\beta$	B	Bcast (EU)	Bcast (3TS)			Bcast (Build)	Bcast (Obtain)				Bcast (4TS)	P2P (9TS)	Bcast (5TS)
		Bz							3 (15TS)					
		C				Bcast (3TS)	P2P (16TS)	4 (Build)			P2P (16TS)			

6.2.1.3 Energy

Energy is a single-objective performance goal, hence only costs variation is applicable. Figure 6.7 shows energy performance of the evolved strategies, presented in Table 6.4, when action and communication costs are varied. We notice that, unlike the Wumpus World, when communication cost is doubled (ac<cc), the amount of communication remains almost the same, although more variant, compared to the case when communication and action costs are equal (ac=cc). Yet, the total amount of energy consumed significantly decreases when (ac<cc), with p=0.03.

Minimization of energy can be achieved by minimizing travel distance and/or minimizing communication. Therefore, a strategy with high communication may outperform another with less communication with respect to energy, if the former enables agents to travel significantly shorter distance. Yet, unlike other multi-objective goals, increasing communication cost does not necessarily mean same or lower communication. Instead, GA may become more restrictive with communication and evolves only what really contributes to reducing travel distance and hence energy. In some cases, communication may also increase when its cost increases, if the evolved strategy reduces travel distance significantly. Energy fitness does not involve assignment of weights to performance goal and communication, which implies uncontrolled communication, and hence GA may accept communication strategies with any communication cost as long as it consumes less energy. This contrasts with other (multi-objective) performance goals, where the fitness balances between the goal performance and communication, according to the assigned weights. Therefore, communication strategies with communication amount that does not comply with its assigned weight are avoided, even if they achieve better goal performance. For this reason, comparing the evolved energy strategies when action and communication costs are varied is ineffectual, yet the evolved strategies with highest fitness in each cost setting are presented in Table 6.4.

In the case of high action cost (ac>cc), the amount of energy considerably reduces, (p=0.0006 and p=0.0007), compared to the cases (ac=cc) and (ac<cc), respectively, and communication significantly increases. Due to the high standard error for communication versus energy in the case (ac>cc), it is believed that GA evolved a large amount of neutral communication, in efforts to reduce energy consumption. The ANOVA results allow us to state with 95% confidence that the amount of energy consumed by the three communication strategies are significantly different (p=0.0003).

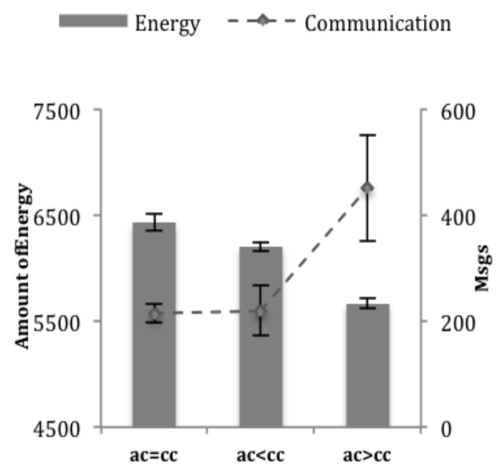


Figure 6.7: Performance of energy strategies.

Table 6.4: The evolved Energy communication strategies.

Fitness Parameters		Information Instances (IS_i)											
Costs		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
ac=cc	B		Bcast (5TS)								Bcast (4TS)		Bcast (3TS)
	Bz							2 (8TS)					
	C			P2P (Build)	3 (5TS)		3 (18TS)					P2P (19TS)	
ac<cc	B	P2P (16TS)	Bcast (EU)			P2P (4TS)	P2P (12TS)				Bcast (3TS)	2 (13TS)	Bcast (3TS)
	Bz							4 (17TS)					
	C				Bcast (EU)	P2P (16TS)	P2P (16TS)						
ac>cc	B	Bcast (Build)	Bcast (Find)				P2P (17TS)	P2P (Find)			Bcast (10TS)	2 (EU)	Bcast (13TS)
	Bz							2 (16TS)					
	C			P2P (9TS)	Bcast (3TS)	3 (10TS)	4 (17TS)				P2P (10TS)		

6.2.1.4 Effort Duplication

Similar to the Wumpus World domain, we notice the popularity of P2P as a recipient strategy in Table 6.5, to prevent informing two agents about the same information, and hence avoid triggering the same decision/action. This is negated in the case of communicating structure updates (IS_2), which is always broadcast, to keep builders continuously updated with the progress of the task, and hence avoid targeting already occupied structure sites.

Within each cost setting, in Figure 6.8, we observe that improvement in performance as a result of goal's weight increase is significant, confirmed by ANOVA test with (p=0.00001 and p=0.008) for (ac=cc and ac>cc), respectively. However, in the cost setting (ac<cc), the effort duplication decreases significantly only when the goal's weight increases from α=0.25 to α=0.5, (p=0.0001), as it maintains the same performance when the goal's weight further increases to reach α=0.75 (p=0.1). The reason is that due to the high communication cost, GA favors a strategy with less communication and higher duplication of effort, since reducing the effort duplication requires high amount of communication that given the goal's weight and communication cost would result in worse fitness.

Similar to time, we observe close performance, in Figure 6.8, as well as strategies, in Table 6.5, for the corresponding strategies when (ac=cc) and (ac>cc) because increasing action cost does not affect effort duplication fitness. In addition, when communication cost is high (ac<cc), GA evolves strategies with less communication, compared to the corresponding strategies when (ac=cc) and (ac>cc), and hence agents duplicate each other's efforts more often, as (p=0.00001) when (α<β), and (p=0.00001) when (α>β). Yet, this does not hold true when communication and effort duplication have equal weights (α=β), as explained previously, because increasing communication cost did not affect the evolved strategy and performance (p=0.96), as shown in Table 6.5, where IS_4 is communicated to the nearest collector, and IS_7 to the closest bulldozer.

When both weight and cost of communication are high (α<β, ac<cc), GA evolved no communication strategy, as shown in Table 6.5. The corresponding strategies when (ac=cc) and (ac>cc) communicate IS_7 to bulldozers as P2P. Although these strategies reduce the effort duplication approximately by half, compared to the no communication strategy, it is believed that the information instance have no direct contribution to the reduction of effort duplication. As explained previously, communicating IS_7 to bulldozers enable them to clear debris from the structure site quickly, and hence soon a block will be attached to complete the structure. Evidently, this information instance has significant effect in reducing the time taken to complete the task, as it was communicated in all time strategies, which implicitly reduce duplication of effort.

When goal's weight is increased (α=β), GA further evolved communicating sufficient supply IS_4 to the closest collector in all cost settings. This information instance prevents collectors from finding and collecting unneeded blocks and placing them in their supply zones that builders never visit, and hence reduces duplication of effort (p=0.001), as collectors will not continuously try to place a block in a full supply zone. However, this information instance is not broadcast to all collectors in order to provide them with more supply zone options, and hence preventing them from targeting a small number of zones.

When effort duplication has higher weight than communication (α>β), builders are allowed to broadcast structure updates (IS_2) to each other, but are not allowed to communicate sufficient supply (IS_4). Instead, they communicate insufficient supply (IS_3) to the closest collector. The latter information instance enables collectors to know what blocks are still needed, which together with (IS_2) have significant effect on reducing duplication of efforts (p=0.0002). We conclude that the most influential information instances to effort duplication are IS_7, which is communicated in eight strategies, followed by IS_4, which is communicated in four strategies, and then IS_2 and IS_3 as they are communicated in two strategies.

Table 6.5: The evolved effort duplication communication strategies.

Fitness Parameters			Information Instances (IS_i)											
Costs	Weights		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
ac=cc	$\alpha<\beta$	B												
		Bz							P2P (Build)					
		C												
	$\alpha=\beta$	B												
		Bz							P2P (Build)					
		C				P2P (14TS)								
	$\alpha>\beta$	B		Bcast (4TS)										
		Bz							P2P (Build)					
		C			P2P (2TS)									
ac<cc	$\alpha<\beta$	B												
		Bz												
		C												
	$\alpha=\beta$	B												
		Bz							P2P (Build)					
		C				P2P (10TS)								
	$\alpha>\beta$	B												
		Bz							P2P (Build)					
		C				P2P (19TS)								
ac>cc	$\alpha<\beta$	B												
		Bz							P2P (18TS)					
		C												
	$\alpha=\beta$	B												
		Bz							P2P (Build)					
		C				P2P (EU)								
	$\alpha>\beta$	B		Bcast (10TS)										
		Bz							P2P (15TS)					
		C			P2P (EU)									

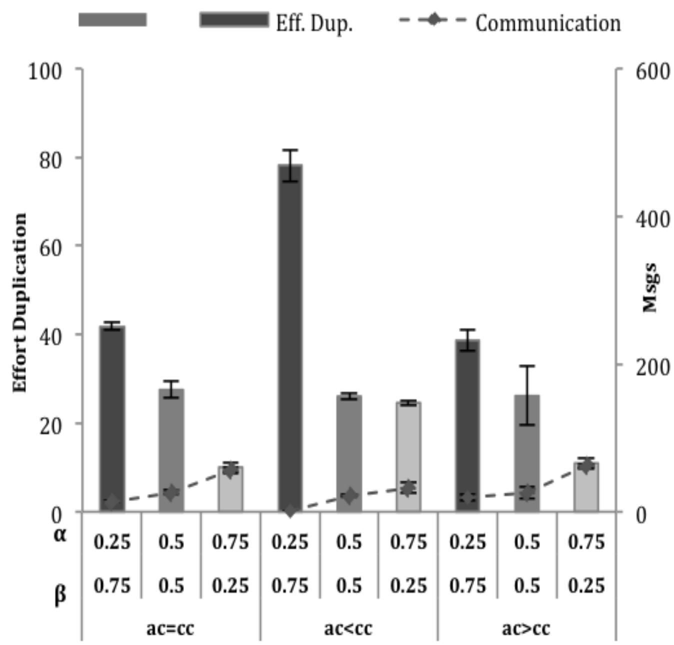


Figure 6.8: Performance of effort duplication strategies.

6.2.2 Fitness Goal

6.2.2.1 $\alpha > \beta$

In this weight setting, the system's designer is interested in improving the goal performance of the system more than minimizing communication. Therefore, a small improvement in the goal performance, attained by a greater increase in communication is accepted, as a consequence of the relative weight of goal performance versus communication. In this domain, and similar to the Wumpus World domain, minimization of time and minimization of travel distance are not always conflicting goals. We shall explore in this and forthcoming sections the cases where time and travel distance conflict and the cases where they do not.

Figures 6.9 and 6.10 compare the time and travel performance, respectively, between strategies of varying goals. When communication cost is high, (Figures 6.9b and 6.10b), we observe similar time performance (p=0.40) and fitness (p=0.49) between the time and travel strategies, yet improved travel performance (p=0.02) and fitness (p=0.01) is achieved by the travel strategy. By comparing the two strategies, in Tables 6.2 and 6.3, we notice that the travel strategy communicates all information instances that are essential to achieving the corresponding time performance, namely, IS_2, IS_4, IS_7, and IS_{12}, hence the close time performance. However, we also notice that the time strategy communicates IS_3, which was avoided in all travel strategies, and also the travel strategy broadcasts IS_4 to all collectors, compared to two collectors in the time strategy. First, we believe that communicating IS_3 may result in increasing the distance traveled by collectors as when collectors receive requests for insufficient supply, they would focus on finding blocks of the color they received insufficient supply for, and hence ignore any block of other colors that they come across. Although this may, in some cases, allow for faster completion of the structure, it costs collectors long distances of travel. Second, broadcasting IS_4 to collectors help decrease the distance traveled by them as they all would focus on finding blocks needed for completing the structure. The energy strategy, although communicates more time and travel favorable information instances, performs as well as the time strategy and travel strategy with respect to time (p=0.22) and travel distance (p=0.06), respectively. The reason is due to the previously reported communication's large standard error for the energy strategy. As explained earlier, energy fitness does not consider weights for goal and communication. Therefore, strategies with less energy consumption are considered better, despite their communication cost. This uncontrolled communication feature of the energy fitness produces highly variant communication strategies that, while consume close amount of energy, produce diverse behavior and performance with respect to other performance goals. Since the effort duplication strategy does not communicates IS_2 and IS_{12} to builders, it does not perform favorably with respect time (p=0.005), and since it further does not broadcast IS_4, it makes agents travel significantly long distance (p=0.0003), compared to the travel strategy.

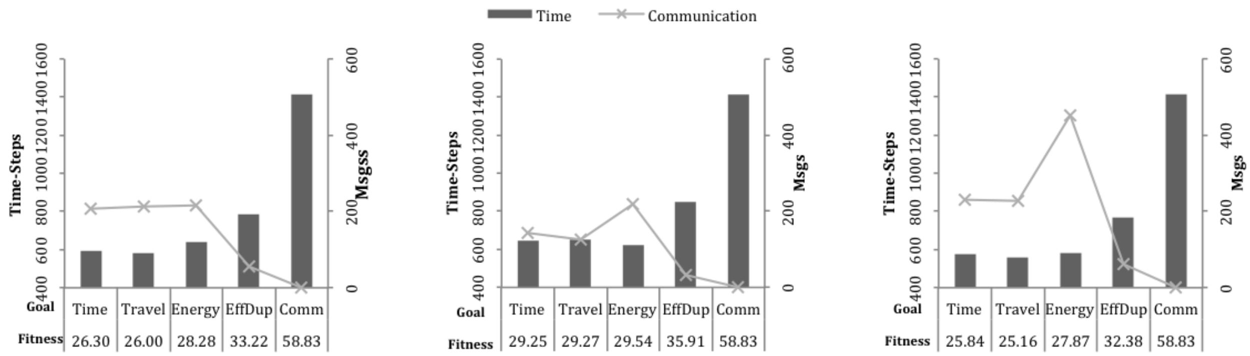
When action and communication costs are equal, and similar to the previous cost setting, time and travel strategies have close time performance (p=0.09) and time fitness (p=0.18), while the latter strategy (as it communicates IS_9) outperforms the former (which communicates IS_3) with respect to travel performance (p=0.01) and travel fitness (p=0.02). The travel strategy performs as well as the time strategy with respect to time because it communicates all important information instances to time, namely, IS_2, IS_4, IS_7, IS_{10}, and IS_{12}, using close recipient and timing strategies. The travel strategy also outperforms the energy strategy with respect to travel distance (p=0.0002) and travel fitness (p=0.001), as the latter does not broadcast IS_4 and does not communicate IS_9. The energy strategy is outperformed by the time strategy with respect to time (p=0.03) and time fitness (p=0.04), since energy strategy communicates full supply zone (IS_{11}) to collectors, which was avoided in all time strategies. The reason is that when every collector shares with the closest one the news that a supply zone is full, the recipient collector will avoid finding and collecting blocks that belong to the supply zone for a hundred time-steps, which may delay the progress of the building task. The effort duplication takes significantly longer time to complete the task, compared to the time strategy (p=0.0004), and also travels longer distance (p=0.0007), compared to the travel strategy, as it does not allow communication of IS_4 and IS_9 to collectors, and IS_{10} and IS_{12} to builders.

When action cost is high, the travel strategy performs as well as the time strategy with respect to time (p=0.11) and time fitness (p=0.11), as it communicates the most influential (favorable) information instances to reduce the time taken to complete the task. In addition, the travel strategy outperforms the time strategy with respect to travel (p=0.006) and travel fitness (p=0.008) for four reasons. The first reason is that the time strategy communicates IS_3, which is believed to increase the distance traveled by collectors. The other three reasons are associated with the travel strategy, which are: (1) it broadcasts IS_4 to collectors, (2) it communicates IS_9, and (3) it communicates IS_1. The energy strategy has close time performance (p=0.32) and travel performance (p=0.25) to the time and travel strategies, respectively, for the same aforementioned reason. However, due to the high amount of communication, it has significantly worse time fitness (p=0.017) and travel fitness (p=0.01).

Figure 6.11 depicts the performance of strategies of varying goals with respect to energy. When communication cost is high (ac<cc), the energy strategy outperforms the time strategy (p=0.003) as the latter does not communicate IS_{10} to builders. Yet, the latter performs as well as the travel strategy (p=0.13), with respect to energy. Although the travel strategy does not communicate IS_{10},

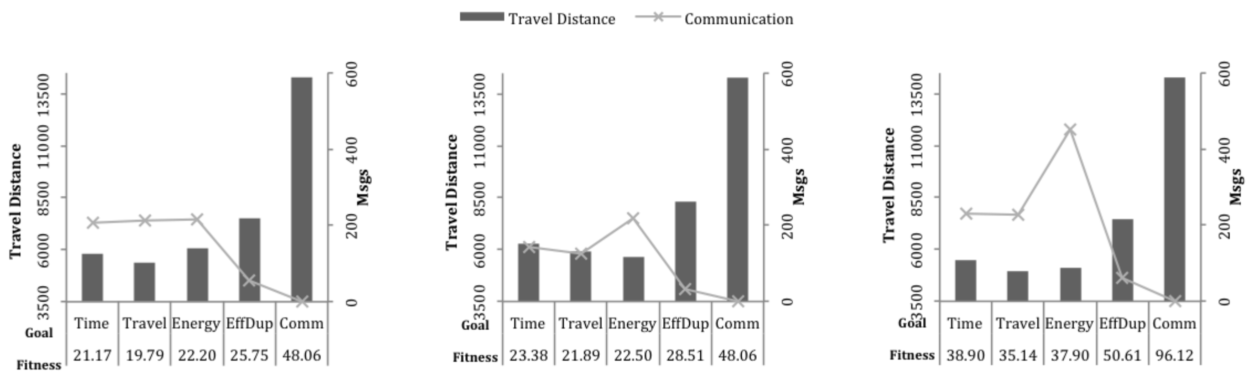
it broadcasts IS_4 to collectors. Due to the high communication cost incurred by the energy strategy, it achieves close energy fitness to time (p=0.12) and travel (p=0.11) strategies. When action and communication costs are equal, the energy strategy has close energy performance (p=0.05) and fitness (p=0.1) to the time strategy, but is outperformed by the travel strategy with respect to energy (p=0.0004) and fitness (p=0.001), as the latter communicates IS_9. When the action cost is high, and similar to the case (ac=cc), the energy strategy has close energy performance (p=0.13) and fitness (p=0.18) to the time strategy, but is outperformed by the travel strategy with respect to energy (p=0.02) and fitness (p=0.01), as the latter communicates IS_9.

The effort duplication performance of different strategies is presented in Figure 6.12. The effort duplication strategy outperforms all other strategies in all cost settings, with respect to effort duplication and fitness, as they communicate more information instances, such as IS_5, IS_6, and IS_{10}, or same information instance but to more recipients, such as IS_4.



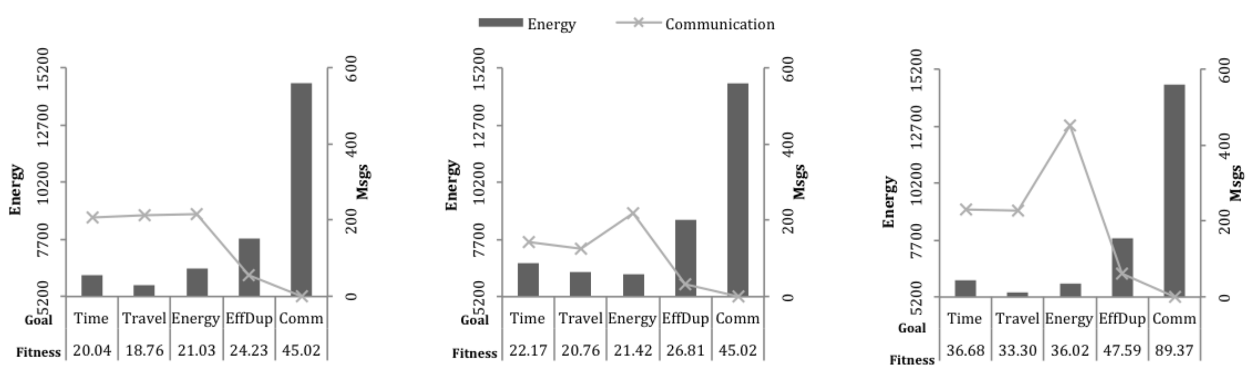
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.9: Time performance of communication strategies of various goals ($\alpha > \beta$)



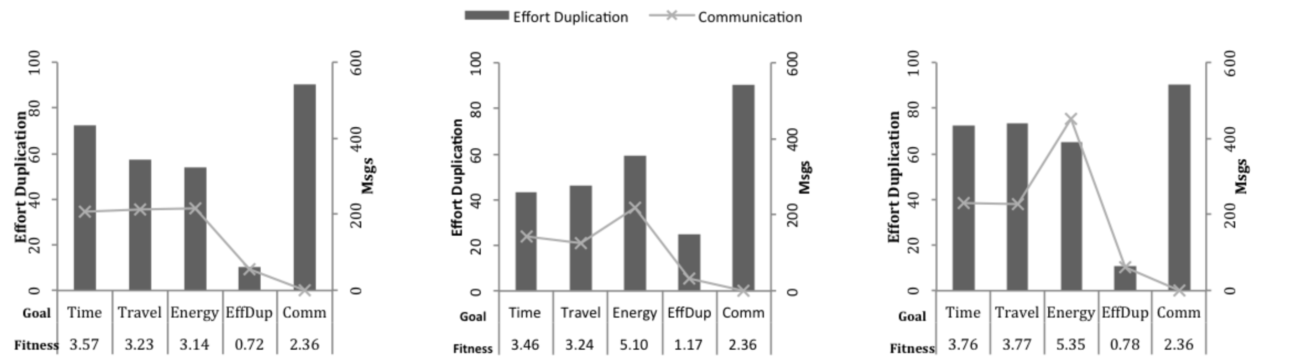
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.10: Travel performance of communication strategies of various goals ($\alpha > \beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.11: Energy performance of communication strategies of various goals ($\alpha > \beta$)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.12: Effort duplication performance of communication strategies of various goals (α>β)

6.2.2.2 α=β

In this case, goal and communication costs contribute equally to the total strategy's fitness. As a result, GA approves increasing communication cost only if it improves goal performance for the same, or close, proportion.

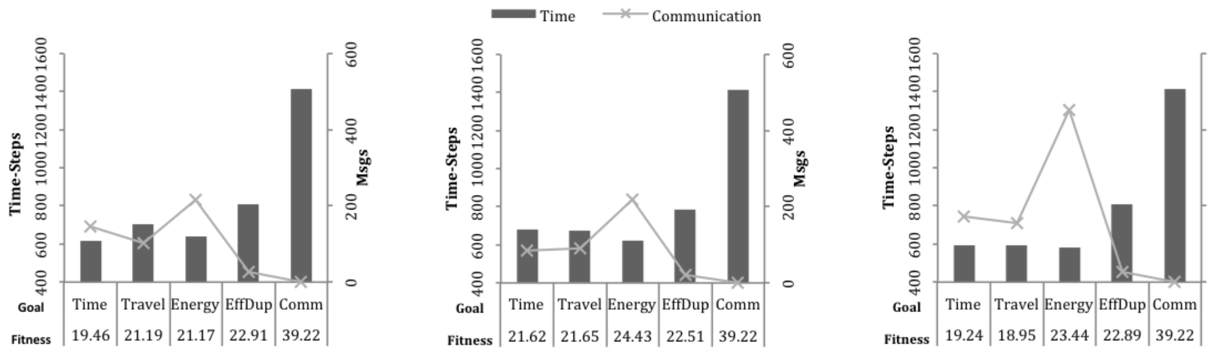
Figure 6.13 depicts the time performance of the strategies. When action and communication costs are equal, the time strategy outperforms the travel strategy with respect to time (p=0.04) and performs as well as the travel strategy with respect to travel distance (p=0.24). The reason for the variant time performance is that the time strategy communicates a time-favorable information instance (IS_{10}) to all builders, which is not communicated by the travel strategy. Yet, the latter communicates instead travel-favorable information instances, namely IS_1 and broadcast IS_4, hence the close travel performance. When it comes to the fitness of the two strategies, the opposite holds true, due to the high communication incurred by the time strategy. Therefore, the travel strategy has close time fitness (p=0.09) and significantly better travel fitness (p=0.01), compared to the time strategy. As the energy strategy communicates all necessary information instances to both performance goals, it performs comparably, with respect to time (p=0.19) and travel distance (p=0.31), to the time and travel distance strategies. Due to its high amount of communication, it has significantly worse time (p=0.04) and travel fitness (p=0.005).

When the communication cost is high, both time and travel strategies perform closely with respect to time (p=0.3) and time fitness (p=0.5), as the latter communicates all information instances that are communicated by the former, yet the latter outperforms the former with respect to travel distance (p=0.003) and travel fitness (p=0.01), as it broadcasts IS_4 to collectors, as oppose to three collectors in the time strategy. The energy strategy is able to complete the task faster than the time strategy (p=0.04) and with travel distance shorter than that of the travel strategy (p=0.02). The reason is that it allows communication of time and travel favorable information instances, which were disallowed in the time and travel strategies due to communication restriction (β=0.5), such as IS_{10} and IS_1. Since the energy strategy communicates larger number of messages, it achieves close time (p=0.1) and worse travel (p=0.03) fitness values.

When action cost is high, time and travel strategies have close time performance (p=0.49) and fitness (p=0.29), as the travel strategy, similar to the previous case, communicates all necessary information instances to reduce time to a comparable performance to the time strategy. However, the travel strategy excels with respect to travel distance (p=0.019) and travel fitness (p=0.013) because the time strategy communicates IS_3 and the travel strategy broadcasts IS_4 to all collectors. The energy strategy performs closely with respect to time (p=0.20) and travel distance (p=0.1), compared to the the time and travel strategies, respectively. However, since it communicates significantly large number of messages, it has significantly unfavorable time (p=0.02) and travel fitness (p=0.02).

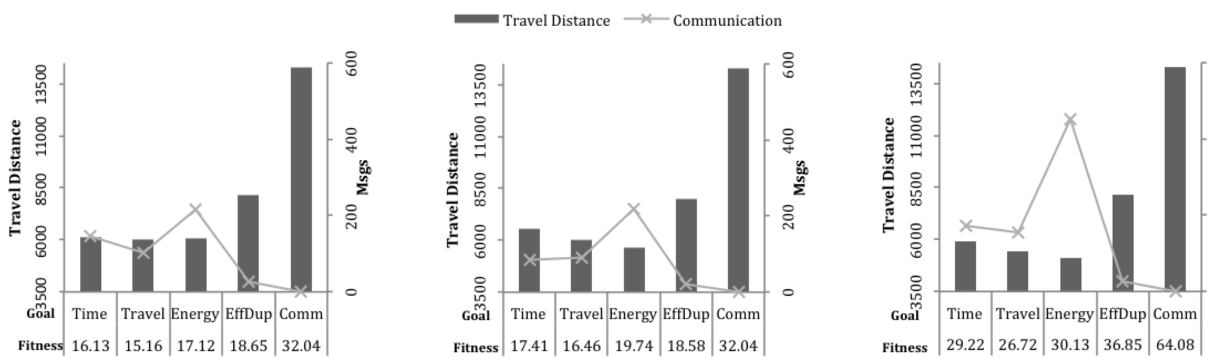
Figure 6.15 shows the energy performance of strategies of different performance goals in three different cost settings. In the case of (ac=cc), both time and travel strategies have comparable energy performance (p=0.42 and p=0.11, respectively) to that of the energy strategy. Since they both communicate less than the energy strategy, they achieve better energy fitness than the energy strategy (p=0.04 and p=0.01), for the time and travel strategies, respectively. When communication cost is high, the energy strategy outperforms the time strategy with respect to energy (p=0.0004), but performs closely to the travel strategy (p=0.08), as it broadcasts IS_4 to collectors. When action cost is high, the energy strategy performs significantly better than the time energy (p=0.004) but close to the travel strategy (p=0.49) with respect to energy for the same aforementioned reason.

The effort duplication performance for strategies of different goals is depicted in Figure 6.16. In the case of equal action and communication costs, the travel strategy performs as well as the effort duplication strategy (p=0.46), as the former communicates IS_1. The latter outperforms all other strategies. When communication cost is high, the time and travel strategies perform closely to the effort duplication strategy (p=0.11 and p=0.25, respectively), although the former two strategies communicate significantly higher. As Tables 6.2 and 6.3 indicate, both strategies communicate IS_2, a duplication-effort-favorable strategy, and S_{11}, which speeds up completion of the building task as bulldozers notify builders once a structure site is cleared, yet they did not outperform the effort duplication strategy. The reason is that both strategies communicate sufficient supply zone (IS_4) to three or more collectors, which increase duplication of efforts as collectors will have fewer choices of supply zones, and hence continuously try to deposit blocks in full zones. It is believed that the pros and cons of the time and travel strategies on one side and the effort duplication strategy on the other side even out the effect of these practices, and hence the close performance. When action cost is high, the effort duplication strategy outperforms all other strategies.



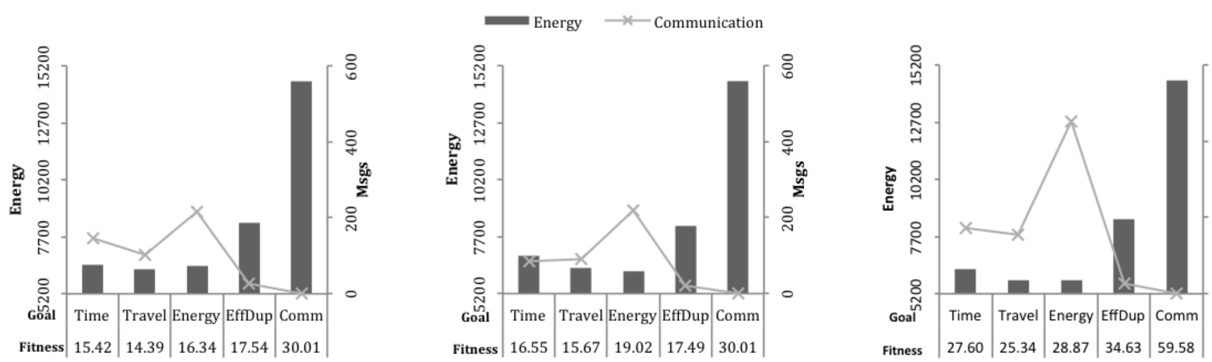
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.13: Time performance of communication strategies of various goals (α=β)



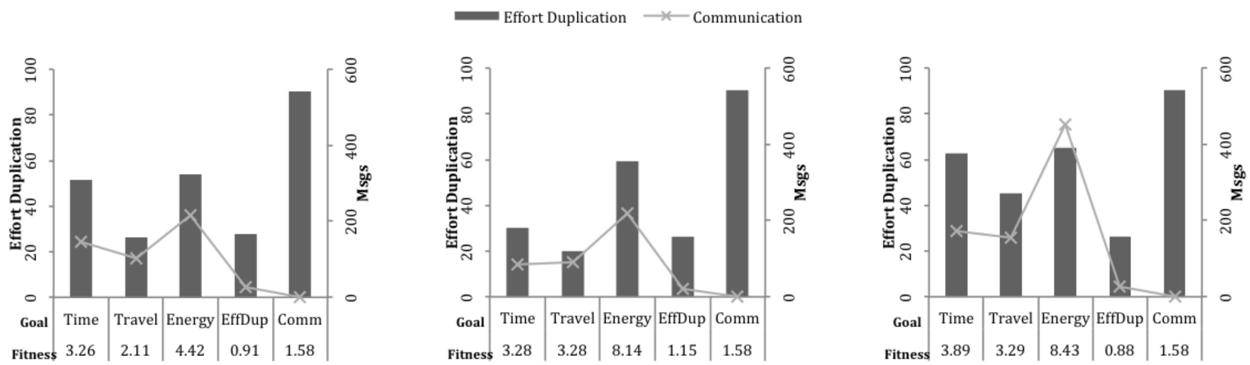
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.14: Travel performance of communication strategies of various goals (α=β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.15: Energy performance of communication strategies of various goals (α=β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.16: Effort duplication performance of communication strategies of various goals (α=β)

6.2.2.3 α<β

This parameter setting is usually used when communication is costly, and hence minimizing it has high priority. At the same time, the system designer is interested in whether low communication can improve the system's performance. Therefore, the aim is to improve the goal performance, but with minimum cost of communication.

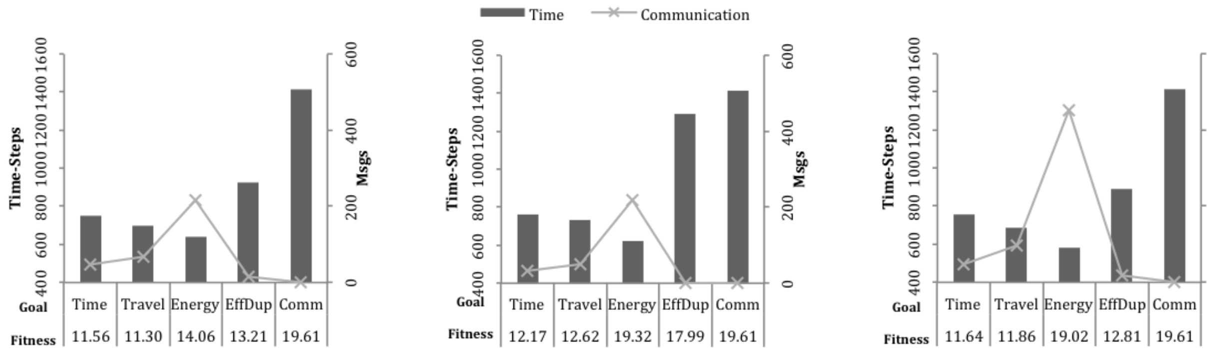
The time and travel performance of the strategies are displayed in Figures 6.17 and 6.18, respectively. Although, when (ac=cc), the travel strategy allows completing the task faster than the time strategy (p=0.02), the time fitness values for both strategies are close (p=0.21). The reason is that the travel strategy communicates more as it allows sending IS_{12}, a time-favorable information instance. Since IS_{12} is also a travel-favorable information instance, besides broadcasting IS_4 to collectors, the travel strategy outperforms the time strategy with respect to travel distance (p=0.006), yet due to the high communication (and high communication weight), the strategies have close travel fitness (p=0.07). As the energy strategy does not consider parameter setting, it communicates more time-favorable information instances, including IS_2, IS_{10}, and and IS_{12}, and hence outperforms the time strategy with respect to time (p=0.005). For the same reason, the time strategy has better time fitness than the energy strategy (p=0.009). However, since the energy strategy does not broadcast IS_4 to collectors, it does not outperform the travel strategy with respect to travel distance (p=0.1), yet the latter has significantly better travel fitness (p=0.002) due to the energy strategy's high communication.

When the communication cost is high (ac<cc), both time and travel strategies communicate only IS_7 and IS_4, although the travel strategy broadcasts the latter information instance. Therefore, the time and travel strategies have close time performance ($p=0.05$), and the latter has significantly better travel performance ($p=0.02$). Due to the higher communication incurred by the travel strategy, the travel fitness values for both strategies are close ($p=0.11$), but the time strategy achieves higher time fitness ($p=0.03$). As the travel strategy communicates only two information instances, and although the energy strategy does not broadcasts IS_4, the energy strategy outperforms both the time ($p=0.002$) and travel ($p=0.001$) strategies with respect to time and travel distance, respectively. Yet, due to the energy strategy's high communication, both strategies have better goal fitness, with ($p=0.02$) for travel distance, and ($p=0.02$) for time.

When action cost is high, the travel strategy increases communication, in efforts to reduce the total distance traveled by agents. Therefore, we observe in Table 6.3 that the travel strategy communicates IS_2, IS_4, IS_6, IS_7, and IS_{12}, as oppose to the time strategy that only communicates IS_4 and IS_7. For this reason, the travel strategy outperforms the time strategy with respect to both time ($p=0.002$) and travel distance ($p=0.0009$). The travel strategy also has better travel fitness ($p=0.006$), but the time fitness values for both strategies are close ($p=0.09$). Due to the absence of communication weight in case of energy strategy, and hence the high communication, the energy strategy outperforms both time ($p=0.0002$) and travel strategies ($p=0.003$) with respect to their goals. However, the time and travel strategies have better time ($p=0.02$) and travel ($p=0.02$) fitness values, respectively.

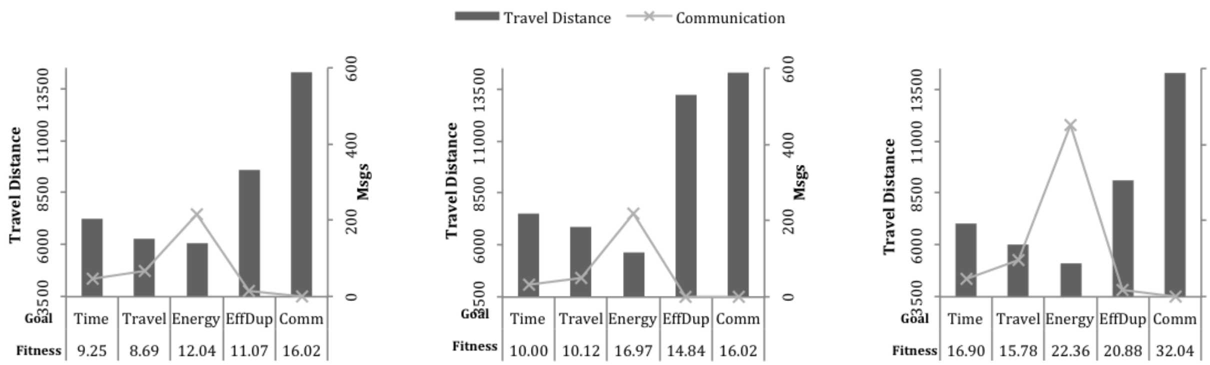
The energy performance of the strategies is shown in Figure 6.19. The comparison between the energy and travel strategies with respect to energy performance resembles that of the travel performance. Both strategies have close energy performance when action and communication costs are equal ($p=0.33$), while the energy strategy has better performance when communication cost is high ($p=0.002$) and when action cost is high ($p=0.004$). The former outperforms the time strategy in all cost settings ($p=0.006$, $p=0.0004$, and $p=0.004$) when (ac=cc), (ac<cc), and (ac>cc). As communication has high weight in this parameter setting, which is not considered by the energy strategy, and hence the high communication, the energy fitness of the energy strategy is significantly worse than that of both time and travel strategies in all cost settings.

In Figure 6.20, the effort duplication performance of the strategies is presented. As the effort duplication strategy communicates minimally, it is outperformed by the time and travel strategies in all cost settings. However, due to the strategies' high communication, they are both surpassed by the effort duplication strategy with respect to the effort duplication fitness. In the case (ac<cc), the energy strategy as well outperforms the effort duplication strategy, as the latter avoids communication at all, but since the former incurs high communication, it scores significantly worse fitness (p=0.04), compared to the effort duplication strategy. In the other two cost settings, the effort duplication strategy outperforms the energy strategy with respect to both goal performance and fitness.



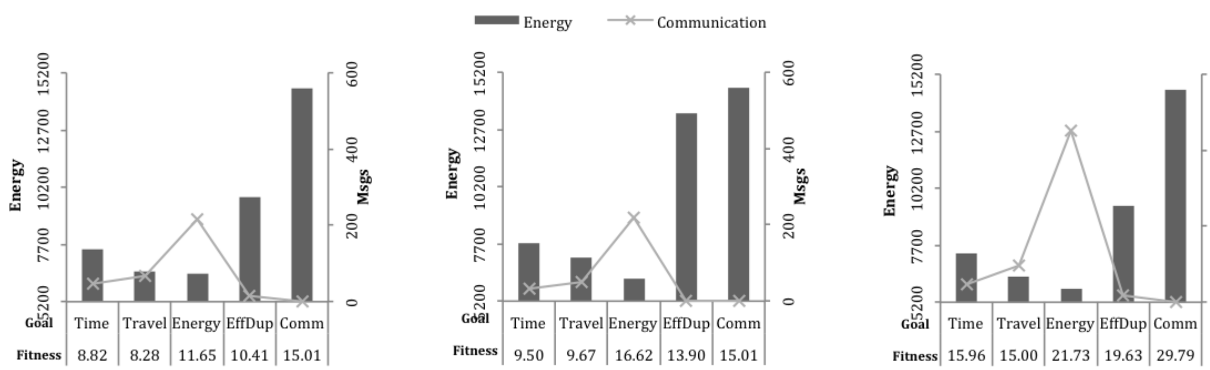
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.17: Time performance of communication strategies of various goals (α<β)



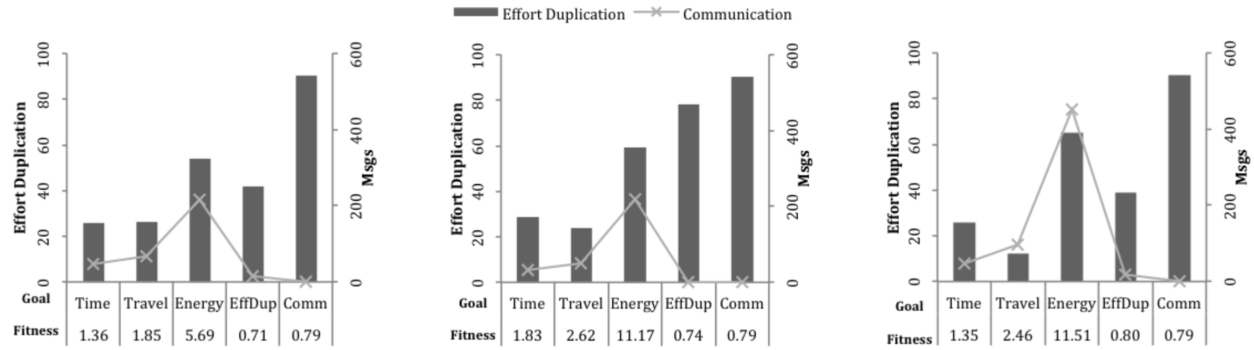
(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.18: Travel performance of communication strategies of various goals (α<β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.19: Energy performance of communication strategies of various goals (α<β)



(a) ac=cc (b) ac<cc (c) ac>cc

Figure 6.20: Effort duplication performance of communication strategies of various goals (α<β)

6.2.3 Simulation Environment

This section provides results for evaluating the proposed approach in difficult scenarios, where the task complexity increases with fixed agent population, and simple scenarios, where agent population increases with fixed task complexity. Similar to the Wumpus World, we seek unbiased communication strategies where no preference is made between goal and communication cost. Therefore, we consider one case of fitness parameters where (α=β, ac=cc) to further investigate the impact of variation of simulation environment on the evolved strategies and goal performance.

6.2.3.1 Agent Population

Agent population in this domain is the total number of builders, bulldozers, and collectors, which is fixed at 13 (three builders, five bulldozers, and five collectors), in the basic setting adapted in all previous experiments in this chapter. In this section, we show the performance of the proposed approach with larger populations.

Figure 6.21 illustrates performance of the evolved strategies, presented in Tables 6.6, 6.7, 6.8, and 6.9, for different populations, with respect to different performance goals. For time, increasing population to double (i.e., 26) allows agents to complete the task in significantly shorter time (p=0.0007), as shown in Figure 6.21a. Yet, the impact of larger population reduces as extra agents make less (p=0.04 for 26 versus 39), and no (p=0.37 for 39 vs 52) difference in the time consumed to finish the task. Communication cost, however, continuoulsy increases with larger population. The evolved strategies, presented in Table 6.6, show consistency with respect to what to communicate. For example, IS_2, IS_4, IS_7, and IS_{12} are communicated in all populations. In addition, and aside from the case of 26 agents, GA always evolves communicating supply zones updates to builders, either by communicating IS_5 or IS_{10}. Further, we notice that the importance of some favorable information instances, such as structure updates (IS_2) and sufficient supply (IS_4), reduces as agent population increases. For example, IS_2 is broadcast to builders in case 13 and 26 agents, yet it is communicated to only a subset when the total population is 39 and 52. The reason is that when the goal is to complete the task in short time, the focus would be on speeding up the progress of building the structure (as oppose to adjusting agents behavior in other performance goals). Therefore, when the number of builders is small (3 and 6), they all need to be updated when a builder attaches a block to a structure site, in order to allow them to use their time effectively by focusing on unoccupied structure sites. However, in case of large number of builders (9 and 12), broadcasting IS_2 is probably unnecessary for quick completion of the task, as only a subset of builders need to focus on attaching blocks to unoccupied sites. Moreover, sufficient supply (IS_4) is communicated to most collectors in case of 13, 26, and 39 agents. However, the number of recipients of this information instance drops to only a small subset of collectors in case of 52 agents. Similar to structure updates (IS_2) in large populations, it is sufficient to inform some collectors about having no longer needed blocks to steer their focus to what is still needed to complete the structure.

Unlike the Wumpus World domain, the total distance traveled by agents continuously increases with larger agent populations, as shown in Figure 6.21b, since bulldozers and collectors walk randomly to find debris and blocks, respectively, hence more agents always result in longer travel distance. The evolved travel strategies, in Table 6.7, show consistency as all information instances that are communicated in the original population, namely, IS_1, IS_2, IS_4, IS_7, and IS_{12}, are communicated in all other populations, albeit changes occurred to the recipient strategies of some information instances. For instance, builder's target IS_1 is only broadcast to builders in the original population case as it is only communicated to a subset in larger populations. It is believed that, since broadcast is a costly recipient strategy, especially in large population, GA restricted recipients of builders' targets (be it structure site or supply zone) to only a subset because builders change their targets frequently. However, builders are still able to broadcast the news that a block has been attached to a site (IS_2) in larger population. This contrasts with the recipient strategy evolved for IS_2 for time strategies, explained earlier, since when the goal is to minimize the total travel distance, the focus would be to make every step of every agent worthwhile. Therefore, no builder should pick up a block and target an already occupied structure site, no matter how many builders there are.

Similar to travel distance, the amount of energy consumed continuously increases with larger populations, yet the amount of communication is kept close in the two largest populations. As energy is a single-objective goal, minimizing energy can be achieved by minimizing communication and/or travel distance. Therefore, in the two largest populations, agents may communicate the same amount of communication, but travel longer distances in the largest population, which results in the high energy consumption.

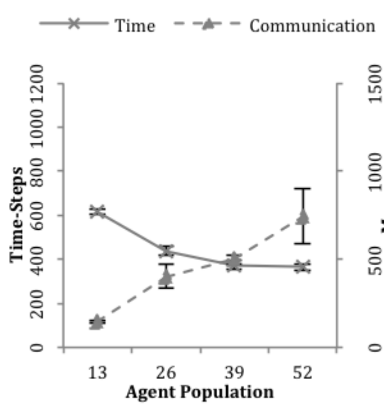
When agent population is doubled, i.e., 26, GA has successfully evolved a strategy to keep duplication of efforts very close (p=0.06) to the previous case of 13 agents. Yet, effort duplication sharply increases afterwards with larger populations. The opposite holds true for communication, as it increases significantly when population is doubled to 26, remains the same with 39 agents, and increases again with the largest population.

The evolved strategies for this performance goal are shown in Table 6.9. We notice that in case of 26 agents, GA evolved a strategy that allows builders to further broadcast structure updates (IS_2) to each other to avoid having a builder that targets an already occupied structure site. In addition, builders are allowed to communicate IS_4 to six collectors, rather than one collector in case of 13 agents. As explained previously, in the original population, communicating IS_4 to many collectors is avoided in order to provide collectors with more supply zones to deposit to, and hence less likely to target a full supply zone. However, in case of larger populations, this no longer holds true. The reason is that larger population means larger number of both builders and collectors. While more builders result in more pick ups from supply zones, and hence less likelihood of full supply zones, more collectors result in the opposite. On one hand, if most collectors are not informed about sufficient supply zones (as in the original population), then they would continuously try to deposit blocks to full supply zones. On the other hand, if all collectors are informed about sufficient supply zones, then they all would focus on a small number of zones, which will make blocks deposit larger than blocks pick up, and hence more likely to deposit to full supply zones. Therefore, we observe that IS_4 is communicated to a subset of collectors in larger populations.

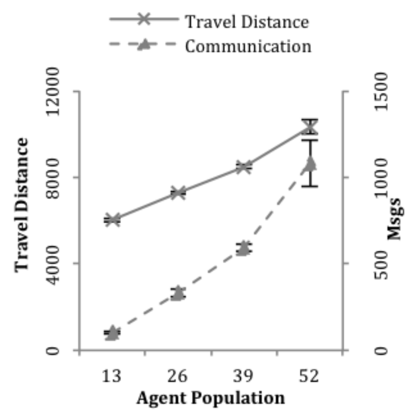
In the case of 39 agents, builders are allowed to share their targets (IS_1), rather than structure updates (IS_2), and further allowed to communicate insufficient supply (IS_3). When builders are allowed to communicate IS_1, they send each other the supply zone or structure site they are targeting. Communicating IS_3 is believed to speed up the building task. In the largest population (52), builders are allowed to communicate both IS_1 and IS_2, in addition to IS_4, IS_7, and IS_{12}. Similar to IS_3, the latter information instance helps builders complete the task faster, as bulldozers inform builders about clear structure sites.

Table 6.6: The evolved time communication strategies for larger populations.

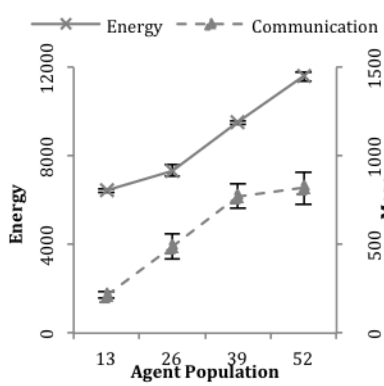
Agent Population				Information Instances (IS_i)											
B Total	Bz	C		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
3 13	5	5	B		Bcast (14TS)								Bcast (Wander)		Bcast (10TS)
			Bz							2 (14TS)					
			C				3 (7TS)						2 (Wander)		
6 26	10	10	B		Bcast (12TS)										3 (15TS)
			Bz							7 (Build)					
			C				9 (Find)								
9 39	15	15	B		2 (4 TS)			7 (EU)							Bcast (7TS)
			Bz							6 (20TS)					
			C				13 (3TS)								
12 52	20	20	B		6 (14TS)				9 (12TS)				Bcast (16TS)		Bcast (7TS)
			Bz							10 (Build)					9 (12TS)
			C				5 (14TS)								



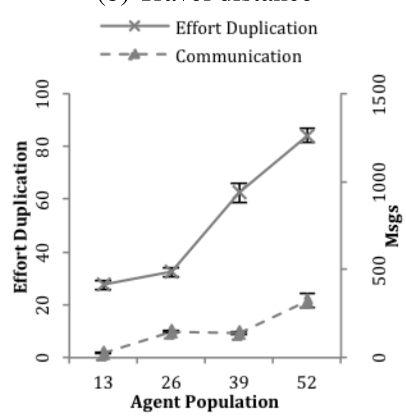
(a) Time



(b) Travel distance



(c) Energy



(d) Effort duplication

Figure 6.21: Various-goals performance with increasing agent population.

Table 6.7: The evolved travel communication strategies for larger populations.

Agent Population				Information Instances (IS_i)											
B Total	Bz	C		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
3 13	5	5	B	Bcast (Find)	Bcast (2TS)										Bcast (4TS)
			Bz							P2P (4TS)					
			C				Bcast (3TS)								
6 26	10	10	B	4 (20TS)	Bcast (18TS)										2 (10TS)
			Bz							3 (9TS)					
			C			5 (Build)	7 (9TS)								
9 39	15	15	B	2 (Find)	Bcast (7TS)										8 (12TS)
			Bz							4 (9TS)					
			C				12 (6TS)		9 (Find)						
12 52	20	20	B	7 (8TS)	Bcast (12TS)										P2P (2TS)
			Bz							6 (5TS)					
			C				19 (3TS)		8 (2TS)				P2P (14TS)		

Table 6.8: The evolved energy communication strategies for larger populations.

Agent Population				Information Instances (IS_i)											
B Total	Bz	C		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
3 13	5	5	B		Bcast (5TS)								Bcast (4TS)		Bcast (3TS)
			Bz							2 (8TS)					
			C			P2P (Build)	3 (5TS)		3 (18TS)					P2P (19TS)	
6 26	10	10	B		Bcast (2TS)			P2P (EU)	3 (Find)	2 (16TS)			Bcast (EU)		3 (14TS)
			Bz							3 (8TS)					
			C				9 (EU)		2 (Deposit)						
9 39	15	15	B		Bcast (15TS)				6 (7TS)	3 (12TS)					4 (EU)
			Bz							7 (3TS)					
			C				Bcast (20TS)	P2P (13TS)	P2P (20TS)						
12 52	20	20	B		Bcast (6TS)				P2P (18TS)	8 (11TS)					2 (2TS)
			Bz							4 (6TS)					
			C			12 (Build)	15 (12TS)	P2P (7TS)	3 (8TS)					14 (Wander)	

Table 6.9: The evolved effort duplication communication strategies for larger populations.

Agent Population				Information Instances (IS_i)											
B Total	Bz	C		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
3 13	5	5	B												
			Bz							P2P (Build)					
			C				P2P (14TS)								
6 26	10	10	B		Bcast (5TS)										
			Bz							P2P (15TS)					
			C				6 (Find)								
9 39	15	15	B	P2P (Find)											
			Bz							3 (Build)					
			C			5 (Obtain)	4 (EU)								
12 52	20	20	B	10 (Find)	6 (10TS)										3 (2TS)
			Bz							3 (5TS)					
			C				8 (4TS)								

6.2.3.2 Task Complexity

In this domain, task complexity refers to: (1) number of blocks available in the environment, (2) amount of debris, and (3) structure size. These three features of the environment were fixed in all previous experiments to 310 (200 blocks, 100 debris, and 10 structure size). The performance of the system with respect to different performance goals and with increasing task complexity is depicted in Figure 6.22. In addition, the evolved communication strategies are presented in Tables 6.10, 6.11, 6.12, and 6.13.

As the assigned task increases in complexity, with fixed population, the time needed to complete the task inevitably increases (Figure 6.22a). Yet, GA evolved a strategy with close communication cost to the cases with less task complexity. Table 6.10 reveals that the evolved strategies for all complexity cases are mostly consistent, as they all mutually communicate IS_2, IS_4, IS_7, IS_{10}, and IS_{12}. Further, we notice that in more complex tasks, where sufficient supply (IS_4) is not broadcast, builders are allowed to communicate insufficient supply (IS_3) to the closest builder. As explained previously, communicating this information instance can reduce the time needed to complete the task, especially in cases when only one or a few structure sites left and no blocks exit in the corresponding supply zones. Another information instance that is communicated in more complex tasks is empty supply zone (IS_6) to builder and/ore collectors. In case of bigger structures, builders need longer time to complete the task, hence informing them that a supply zone is empty can help them use their time more effectively. Also, communicating the same information instance to collectors encourages them to deposit more blocks in case they are still needed.

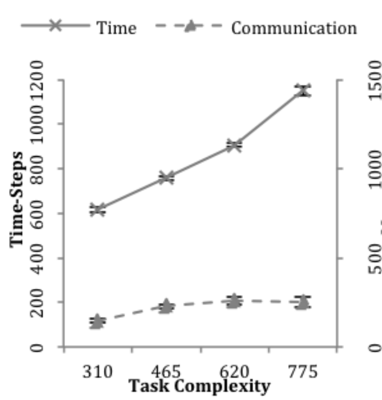
The trend for the travel performance of strategies with more complex tasks is similar to that of the time performance, as illustrated in Figures 6.22b and 6.22a, respectively. There are multiple changes that the travel strategies go through as the task complexity increases. First, builder's target (IS_1) is only communicated in the original case. As the task gets more complex, including bigger structure size, builders do not need to avoid each others targets (such as structure site and supply zone), as it is less likely that two or more builders would target the same structure site, and if that happens, it is more likely that they will find a close structure size that need the block that they carry. Second, we notice that insufficient supply (IS_3) is allowed to be communicated in more complex scenarios. The reason is that as the task gets more complex, collectors need to concentrate on finding blocks needed for the completion of the structure. Third, in the second and third scenarios, builders are kept informed about the status of the supply zones, by communicating IS_5, and IS_{10} or IS_{11}. In the most complex scenario, builders are only informed when a supply zone is full (IS_{11}). It is believed that the reason is that since the importance of travel equals that of communication, with continuous increase of travel distance, GA may try to decrease communication to improve fitness.

Figure 6.22c shows the energy performance of the energy strategies as the task gets more complex. We notice that the energy consumed in each case is very close to the corresponding distance traveled in Figure 6.22b. The reason is due to the high communication in energy strategies, which enables further decrease in the travel distance, and hence less energy consumption. We observe a sudden decrease in communication in the most complex scenario. As explained earlier, the amount of energy consumed can be reduced by reducing travel distance and/or communication. As energy has no weights that control the trade-off between the two objectives, results of energy can be random in this matter.

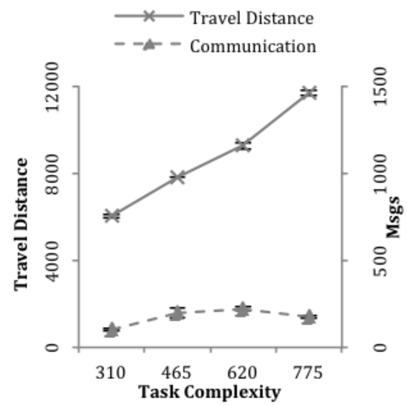
The strategies performance with respect to effort duplication for more complex scenarios is shown in Figure 6.22d. We observe that duplication of efforts decreases with the second complex scenario, which happened for two reasons. First, as the number of available blocks, amount of debris, and structure size increase, it is less likely that two or more collectors target the same block, two or more bulldozers target the same debris, and two or more builders target the same structure site. Second, as shown in Table 6.13, GA found it worthwhile to further communicate structure updates (IS_2) and increase recipients of IS_4, hence resulting in less duplication of efforts. When the task is further complicated, although the first argument still holds true, agents take longer time to complete the task, hence builders and bulldozers may duplicate each others' efforts when they pick up or deposit a block in a supply zone, respectively. We observe that the evolved effort duplication strategies in Table 6.13 for more complex tasks are consistent .

Table 6.10: The evolved time communication strategies for more complex tasks.

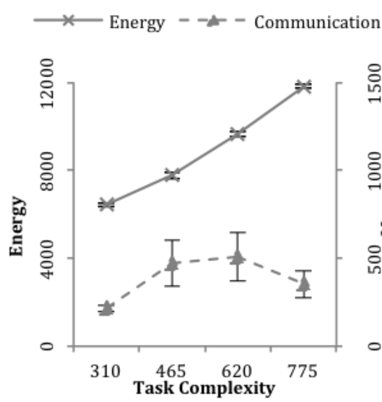
Task Complexity				Information Instances (IS_i)											
Blocks Total	Debris	Structure Size		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
200 310	100	10	B		Bcast (14TS)								Bcast (Wander)		Bcast (10TS)
			Bz							2 (14TS)					
			C				3 (7TS)						2 (Wander)		
300 465	150	15	B		Bcast (3TS)				P2P (3TS)				Bcast (2TS)		P2P (5TS)
			Bz							2 (7TS)					
			C				Bcast (2TS)		P2P (16TS)						
400 620	200	20	B		Bcast (13TS)								Bcast (19TS)		Bcast (5TS)
			Bz							3 (19TS)					
			C			P2P (3TS)	3 (7TS)		P2P (11TS)						
500 775	250	25	B		Bcast (3TS)				P2P (Build)				Bcast (5TS)		P2P (3TS)
			Bz							2 (19TS)					
			C			P2P (5TS)	2 (19TS)								



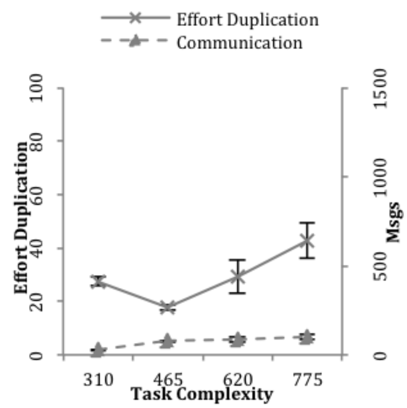
(a) Time



(b) Travel distance



(c) Energy



(d) Effort duplication

Figure 6.22: Various-goals performance with increasing task complexity.

Table 6.11: The evolved travel communication strategies for more complex tasks.

Task Complexity				Information Instances (IS_i)											
Blocks Total	Debris	Structure Size		IS_1	IS_2	IS_3	IS_4	IS_5	IS_6	IS_7	IS_8	IS_9	IS_{10}	IS_{11}	IS_{12}
200 310	100	10	B	Bcast (Find)	Bcast (2TS)										Bcast (4TS)
			Bz							P2P (4TS)					
			C				Bcast (3TS)								
300 465	150	15	B		Bcast (12TS)			P2P (18TS)						P2P (17TS)	Bcast (19TS)
			Bz							P2P (9TS)					
			C			2 (Find)	Bcast (12TS)								
400 620	200	20	B		Bcast (4TS)			P2P (4TS)					P2P (2TS)		2 (6TS)
			Bz							P2P (EU)					
			C			2 (EU)	Bcast (8TS)		P2P (19TS)						
500 775	250	25	B		Bcast (14TS)									P2P (EU)	Bcast (EU)
			Bz							P2P (18TS)					
			C			P2P (3TS)	4 (3TS)								

Table 6.12: The evolved energy communication strategies for more complex tasks.

Task Complexity				Information Instances (IS_i)											
Blocks Total	Debris	Structure Size		IS$_1$	IS$_2$	IS$_3$	IS$_4$	IS$_5$	IS$_6$	IS$_7$	IS$_8$	IS$_9$	IS$_{10}$	IS$_{11}$	IS$_{12}$
200 310	100	10	B		Bcast (5TS)								Bcast (4TS)		Bcast (3TS)
			Bz							2 (8TS)					
			C			P2P (Build)	3 (5TS)		3 (18TS)					P2P (19TS)	
300 465	150	15	B	Bcast (15TS)	Bcast (7TS)			Bcast (9TS)					Bcast (6TS)	2 (5TS)	2 (20TS)
			Bz							P2P (10TS)					
			C			P2P (5TS)	Bcast (4TS)		Bcast (9TS)			3 (7TS)			
400 620	200	20	B	P2P (13TS)	Bcast (11TS)								Bcast (15TS)		Bcast (20TS)
			Bz							P2P (2TS)					
			C			P2P (Find)	Bcast (EU)	2 (14TS)	2 (Find)		P2P (20TS)	3 (8TS)	2 (14TS)	Bcast (7TS)	
500 775	250	25	B		Bcast (9TS)								P2P (9TS)		
			Bz							P2P (12TS)					
			C			P2P (9TS)	4 (8TS)							P2P (17TS)	

Table 6.13: The evolved effort duplication communication strategies for more complex tasks.

Task Complexity				Information Instances (IS_i)											
Blocks Total	Debris	Structure Size		IS$_1$	IS$_2$	IS$_3$	IS$_4$	IS$_5$	IS$_6$	IS$_7$	IS$_8$	IS$_9$	IS$_{10}$	IS$_{11}$	IS$_{12}$
200 310	100	10	B												
			Bz							P2P (Build)					
			C				P2P (14TS)								
300 465	150	15	B		Bcast (Find)										
			Bz							P2P (11TS)					
			C				2 (11TS)		P2P (Build)						
400 620	200	20	B		Bcast (13TS)										
			Bz							P2P (18TS)					
			C				2 (9TS)								
500 775	250	25	B		Bcast (4TS)										
			Bz							P2P (Build)					
			C				2 (11TS)								

6.3 Summary

The Collective Construction is a more complex domain than the Wumpus World, as it includes more information instances and more agent types. Yet, the experimental results, presented in this chapter, support our findings on the Wumpus World domain. We list some of these findings in this section and provide more detailed discussion in the next chapter.

We have demonstrated that, similar to the Wumpus World, information instances in this domain can be classified to favorable, unfavorable, and neutral information instances, with respect to each performance goal, and that specification of each information instance and its impact on goal performance can differ when the simulation environment changes. For example, communicating the information instances IS_2, IS_{10}, and IS_{12} to builders, IS_7 to bulldozers, and IS_3, IS_4, and IS_{10} to collectors are considered favorable information instances to time performance. In addition, communicating IS_1 to builders, IS_9 to collectors, and IS_6 is considered neutral to the time performance, while IS_{11} to collectors is considered unfavorable. In scenarios with larger population, we observed reduction in the favorable impact of communicating IS_2 to builders on time performance, and increase in the favorable impact of communicating IS_7 to bulldozers. In cases with more complex tasks, we observed increase in the favorable impact of communicating IS_3 to collectors and IS_6, and decrease in communicating IS_{10} to collectors.

In the case of travel distance, the favorable information instances include communicating IS_1, IS_2, IS_6, IS_{10}, and IS_{12} to builders, and IS_7 to bulldozers, and IS_4, IS_6, and IS_9 to collectors. Communicating IS_3 is considered unfavorable to the travel performance. We observed reduction in the favorable impact of communicating IS_1 in scenarios with larger populations and scenarios with more complex tasks, while communicating IS_3 to collectors became favorable in scenarios with more complex tasks.

When the goal is to reduce duplication of efforts, the most influential information instances are IS_2 to builders, IS_7 to bulldozers, and IS_3 and IS_4 to collectors. Moreover, we observed increase in the influence of communicating IS_2 to builders and IS_4 to collectors in scenarios with larger populations and scenarios with more complex tasks. Additionally, the favorability of communicating IS_1 to builders increased in scenarios with larger populations.

Furthermore, we have shown that inverse variation between costs and weights of goal and communication can even out their importance, and hence results in similar communication strategies and performance. Besides, our results indicate that the proposed approach works less favorably with single-objective performance goal (energy in this case), due to uncontrolled communication.

CHAPTER 7

Conclusion

7.1 Summary

This dissertation has proposed, examined, and evaluated the validity of the hypothesis that:

> Communication strategies in multi-agent systems, namely what, when, and to whom agents communicate, can be learned in order to improve the performance of the system with respect to flexible performance goals, resulting in learning of goal-driven communication.

In support of this claim, we conducted research to investigate the plausibility of evolving goal-driven communication using Genetic Algorithm. We provided our definition of Communication Strategy, including two components; recipients and timing strategies, and we designed a fitness function that balances between goal performance and communication cost, according to user-defined weights. We applied the proposed approach to two different, well-known, multi-agent case studies, namely, the Wumpus World and Collective Construction, based on thoroughly-designed experiments that consider variations of three different factors. We, further, supported our analysis and evaluation of the obtained results with statistical evidence to validate the effectiveness of the approach from different angles.

Our experimental results indicate that GA has successfully evolved goal-driven multi-agent communication strategies. Our research shows that the system's performance can not only be significantly affected by communication variation, but further highly tuned by controlling communication. The presented approach has been confirmed to possess significant utilization in improving the performance with respect to the desired performance goal and trade-off between performance and communication cost.

Variation of the performance goal with fixed weights has shown that a goal-driven strategy cannot be outperformed with respect to goal-based fitness by strategies of other goals. However, other strategies may outperform a goal-driven strategy with respect to its goal if they communicate more to the extent that results in better performance, but causes worse fitness. Further, evolving single-objective communication strategies, i.e., energy-driven strategies, with the absence of trade-off between performance goal and communication produced less quality results due to over-communication.

Variation of goal's weight, with the same goal, reveals that with a greater value for the weight, our approach is able to produce strategies with same or better performance, compared to others with smaller goal's weight. In the worst case of evolving strategies with same performance, one of two situations could be the case. First, increasing communication does not improve performance, as the best performance that the system can achieve was already evolved with smaller goal's weight, and is evolved again for greater weight. Second, increasing communication can improve performance, but the associated communication cost results in worse fitness, hence GA favors same strategies, evolved for less goal's weight, with worse performance and higher fitness. Variation of features of simulation environment, including agent population and task complexity, indicates that the evolved strategies are bound up with the considered simulation environment, and that the proposed approach may perform less favorably in some cases of simple scenarios, as evolved strategies were less reliable, confirmed by high standard error of communication.

Information instances (*IS*) are classified, with respect to each performance goal, into three categories, favorable *IS*, unfavorable *IS*, and neutral *IS*. Favorable information instances are those that, if communicated, can improve the goal performance, whilst unfavorable ones affect the goal performance negatively. Neutral information have no influence on the goal performance, but only increase communication. Our results show that the favorable impact of some information instances and the class that they belong to may change in response to changes in the simulation environment. In addition, the results indicate that GA has success in evolving strategies that allow communicating favorable information to the stated goal and avoid invalid and unfavorable, and neutral information. Yet, assigning a relatively low weight to communication cost or evolving a strategy for simple scenarios may sometimes result in higher standard error for communication, as GA may evolve a communication strategy that allows communicating neutral information. Therefore, even in case of communication at no cost, it is recommended to assign communication a low, yet comparable, weight with that of the goal.

7.2 Contributions

The work presented in this dissertation has made multiple contributions to the state of the art, which we enumerate below:

1. **We have developed an evolutionary approach that given a performance goal, produces a communication strategy that can improve a multi-agent system's performance with respect to the desired goal**. We have evolved strategies with four different performance goals, and examined, in Sections 5.2.2 and 6.2.2, the impact of the chosen performance goal on the actual evolved communication strategy, the system's performance and the strategy's fitness . Further, we have demonstrated under which conditions a communication strategy with goal P_1 may outperform another with goal P_2, with respect to P_2.

2. **We have demonstrated how our approach provides a tool for customizing the tradeoff between the system's performance and communication cost**. We considered nine cases

of fitness parameters; including goal's and communication's weights and action and communication costs. We presented, in Sections 5.2.1 and 6.2.1, the evolved strategies and result performance with each considered performance goal. We, further, showed under which conditions increasing performance goal's weight does not result in better performance with respect to the desired goal.

3. **We have classified information instances into three categories according to their influence on a performance goal, namely favorable, neutral, and unfavorable information instances**. This classification is based on the cases at which an information instance has been allowed to be communicated for each performance goal and its impact on the result performance with respect to the goal.

4. **We demonstrated how different environments features, such as more complex task or larger populations, may call for different behavior to improve performance, reflected in modifications in the evolved strategies**. We have shown examples of this situation in Sections 5.2.3 and 6.2.3. It is partially due to the fact that the value of an information instance to a goal may vary with different environment features.

5. **We have demonstrated how our approach can assist system designers to figure out the potentially best performance that the system can achieve with respect to a specific goal, such as the minimum time or energy that a task takes to complete**. As previous research confirmed that full communication may hinder performance, finding how well a system can perform with respect to a goal, with no design changes, may become difficult. Therefore, this approach enables a system designer to find out the best performance of a system, and then choose among the performance of the system with multiple communication strategies of varying goals and select the one that has the best fit to the system's needs.

7.3 Limitations And Future Work

There are number of limitations that this work has been subject to. This section lists these limitations, explains their impact on our findings, and provides suggestions on how they could be overcome in future research.

1. Both multi-agent case studies, considered in this research, have static environment, although real-world environments are dynamic. The work presented in this dissertation is considered a proof of concept, in which we try to verify the usefulness and feasibility of learning goal-driven communication in Multi-Agent systems. Therefore, similar to other research of this kind, we test our hypothesis on a simple case, in preparation of moving to more complex scenarios, if successful. In future research, we would like to expand this research to be able to apply it on dynamic environments.

2. In this work, recipients of value updates of an information instance are chosen only based on their distance from the sender. In the future, we plan to incorporate other criteria such as agents in a specific range, or based on their location.

There are several future directions of research to either improve the proposed approach or to further enhance its evaluation:

1. The goal-driven communication strategy can be enhanced to include not only what, when, and to whom agents communicate, but also how agents communicate, such as by a message board, pheromone, or other means.

2. Future work includes applying the approach to different application domains, with more groups of agents and different metrics, such as work progress, quality of solution, and agents' idle time.

3. Utilizing Multi-Objective Genetic Algorithm enables evolving the Pareto front, hence the the system designer may decide which the best communication strategy is.

4. Rather than evolving a single goal-driven strategy, this work can be extended to evolve a separate strategy for each agent.

5. Future work can include evolving strategies for each sub-task that agents perform, and allow them to switch between strategies according to the task that they are currently performing.

Bibliography

Abdelmoumène, H. & Belleili, H. (2015). An extended version of opportunity cost algorithm for communication decisions. *Evolving Systems*, 7, 41–60.

Abdullah, N. N. B. (2005). The induction of communication protocols. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 1381–1381). Utrecht, Netherlands: ACM.

Afsharchi, M. & Far, B. H. (2006). Automated ontology evolution in a multi-agent system. In *Proceedings of the 1st International Conference on Scalable Information Systems* Hong Kong: ACM.

Agah, A. & Bekey, G. A. (1998). Evolutionary communication in distributed multi-agent systems. *Mohammad Jamshidi, Francois Pierrot, and Mohamed Kamel (Eds.) Robotics and Manufacturing Systems, Recent Results in Research, Development, and Applications*, 7, 9–14.

Allen, M., Goldman, C. V., & Zilberstein, S. (2005). Learning to communicate in decentralized systems. In *Proceedings of the Twentieth National Conference on Artificial Intelligence, Workshop on Multiagent Learning* Pittsburgh, Pennsylvania, USA.

Allen, T. J., Lee, D. M. S., & Tushman, M. L. (1980). R&D performance as a function of internal communication, project management, and the nature of the work. *IEEE Transactions on Engineering Management*, EM-27(1), 2–12.

Althnian, A. & Agah, A. (2015). Evolutionary learning of goal-oriented communication strategies in multi-agent systems. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 9(3), 52–64.

Althnian, A. & Agah, A. (2016). Evolving goal-driven multi-agent communication: what, when, and to whom. *Evolutionary Intelligence*, (pp. 1–22).

Arabas, J., Michalewicz, Z., & Mulawka, J. (1994). Gavaps-a genetic algorithm with varying population size. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence* Orlando, FL.

Austin, J. L. & Urmson, J. (1962). *How to Do Things with Words*. Oxford, United Kingdom: Clarendon Press.

Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford, United Kingdom: Oxford university press.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms* (pp. 14–21). Hillsdale, NJ, USA.

Balch, T. & Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 27–52.

Baldwin, T. T., Bedell, M. D., & Johnson, J. L. (1997). The social fabric of a team-based mba program: Network effects on student satisfaction and performance. *Academy of Management Journal*, 40(6), 1369–1397.

Balkundi, P. & Harrison, D. A. (2006). Ties, leaders, and time in teams: Strong inference about network structure's effects on team viability and performance. *Academy of Management Journal*, 49(1), 49–68.

Baray, C. (1997). Evolving cooperation via communication in homogeneous multi-agent systems. In *Proceedings of Intelligent Information Systems* (pp. 204–208). Grand Bahama Island, Bahamas: IEEE.

Barbuceanu, M. & Fox, M. S. (1996). Capturing and modeling coordination knowledge for multi-agent systems. *International Journal of Cooperative Information Systems*, 5, 275–314.

Becker, R., Carlin, A., Lesser, V., & Zilberstein, S. (2009). Analyzing myopic approaches for multi-agent communication. *Computational Intelligence*, 25(1), 31–50.

Becker, R., Zilberstein, S., & Lesser, V. (2004a). Decentralized markov decision processes with event-driven interactions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1 (pp. 302–309). New York, NY, USA: IEEE Computer Society.

Becker, R., Zilberstein, S., Lesser, V., & Goldman, C. V. (2004b). Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, (pp. 423–455).

Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4), 819–840.

Bhandari, D., Murthy, C., & Pal, S. K. (2012). Variance as a stopping criterion for genetic algorithms with elitist model. *Fundamenta Informaticae*, 120(2), 145–164.

Bigham, J. & Du, L. (2003). Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 568–575). Melbourne, VIC, Australia: ACM.

Blum, S., Puisa, R., Riedel, J., & Wintermantel, M. (2001). Adaptive mutation strategies for evolutionary algorithms. In *The Annual Conference: EVEN at Weimarer Optimierungsund Stochastiktage* Germany.

Blute, M. (1979). Learning theory and the evolutionary analogy. Cogprints, `http://cogprints.org/858/Blute.pdf`. Accessed: Aug. 2016.

Bonilla-Huerta, E., Hernández, J. C. H., & Morales-Caporal, R. (2011). A specialized random multi-parent crossover operator embedded into a genetic algorithm for gene selection and classification problems. In *Electrical Engineering and Control* (pp. 559–566). Springer.

Bouslimi, I., Hanachi, C., & Ghedira, K. (2014). An experimental evaluation of communication in an organization-based multi-agent system. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 3 (pp. 72–78). Warsaw, Poland: IEEE.

Braubach, L. & Pokahr, A. (2007). Goal-oriented interaction protocols. In *Multiagent System Technologies* (pp. 85–97). Springer.

Bussink, D. (2004). A comparison of language evolution and communication protocols in multiagent systems. In *1st Twente Student Conference on IT, Track C-Intelligent_Interaction* Enschede, Netherlands: Citeseer.

Cangelosi, A. (1999). Modeling the evolution of communication: From stimulus associations to grounded symbolic associations. *Advances in Artificial Life*, (pp. 654–663).

Cangelosi, A. (2001). Evolution of communication and language using signals, symbols, and words. *IEEE Transactions on Evolutionary Computation*, 5(2), 93–101.

Carlin, A. & Zilberstein, S. (2009). Myopic and non-myopic communication under partial observability. In *International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, volume 2 (pp. 331–338). Milano, Italy: IET.

Caruana, R. A., Eshelman, L. J., & Schaffer, J. D. (1989). Representation and hidden bias ii: Eliminating defining length bias in genetic search via shuffle crossover. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, volume 1 (pp. 750–755). Detroit, MI, USA: Morgan Kaufmann Publishers Inc.

Chakraborty, D. & Sen, S. (2007). Computing effective communication policies in multiagent systems. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems* Honolulu, HI, USA: ACM.

Cheong, C. & Winikoff, M. (2005). Hermes: Designing goal-oriented agent interactions. In *Agent-Oriented Software Engineering VI* (pp. 16–27). Springer.

Chopra, A. K., Artikis, A., Bentahar, J., Colombetti, M., Dignum, F., Fornara, N., Jones, A. J., Singh, M. P., & Yolum, P. (2013). Research directions in agent communication. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(2), 20.

Clair, A. S. & Matarić, M. J. (2011). Modeling action and intention for the production of coordinating communication in human-robot task collaborations. In *Proceedings of the 21st International Symposium on Robot and Human Interactive Communication: Workshop on robot feedback in HRI* (pp. 39–46). Paris, France.

Conforth, M. & Meng, Y. (2008). Communication scheme comparison for a distributed multiagent system. In *Proceedings of the 2008 International Conference on Artificial Intelligence*, volume 2: Nevada, USA.

Darnall, R. W. & Preston, J. M. (2012). Beginning project management. `http://2012books.lardbucket.org/pdfs/beginning-project-management-v1.1.pdf`. Accessed: Aug. 2016.

Darwin, C. (1859). *On the origins of species by means of natural selection*. New York, NY, USA: P F Collier & Son Company.

Davis, L. (1989). Adapting operator probabilities in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 61–69). San Francisco, CA, USA.

De Greeff, J. & Nolfi, S. (2010). Evolution of implicit and explicit communication in mobile robots. In *Evolution of Communication and Language in Embodied Agents* (pp. 179–214). Springer.

De Jong, K. A. & Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1), 1–26.

Devi, S., Jadhav, D. G., & Pattnaik, S. S. (2011). Memetic algorithm and its application to function optimization and noise removal. In *1st World Congress on Information and Communication Technologies (WICT)* (pp. 748–753). Mumbai, India: IEEE.

Diaz-Gomez, P. A. & Hougen, D. F. (2007). Initial population for genetic algorithms: A metric approach. In *Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods* (pp. 43–49). Las Vegas, Nevada, USA.

Dignum, F. (2004). *Advances in Agent Communication: International Workshop on Agent Communication Languages*, volume 2922. Melbourne, Australia: Springer Science & Business Media.

Dignum, F. & Greaves, M. (2000). Issues in agent communication: An introduction. In *Issues in Agent Communication* (pp. 1–16). Springer.

Dowell, M. L. & Stephens, L. M. (2001). Determining improved values for coordination and communication parameters in multiagent systems: A case study. Citeseerx, `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.9602&rep=rep1&type=pdf`, Accessed: Aug. 2016.

Dutta, P. S., Goldman, C. V., & Jennings, N. R. (2007). Communicating effectively in resource-constrained multi-agent systems. In *The 20th International Joint Conference on Artificial Intelligence* (pp. 1269–1274). Hyderabad, India.

Dutta, P. S., Jennings, N. R., & Moreau, L. (2005). Cooperative information sharing to improve distributed learning in multi-agent systems. *Journal of Artificial Intelligence Research*, (pp. 407–463).

Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2), 124–141.

Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2011). Ga with a new multi-parent crossover for solving ieee-cec2011 competition problems. In *2011 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1034–1040). New Orleans, LA, USA: IEEE.

Enee, G., Escazut, C., & Defoin-Platel, M. (2004). Evolution of communication between genetic agents. In *16th European Conference on Artificial Intelligence*, volume 16 (pp. 1005). Valencia, Spain.

Eshelman, L. J., Caruana, R. A., & Schaffer, J. D. (1989). Biases in the crossover landscape. In *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 10–19). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). Kqml as an agent communication language. In *Proceedings of the Third International Conference on Information and knowledge Management* (pp. 456–463). New York, NY, USA: ACM.

Floreano, D., Mitri, S., Magnenat, S., & Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. *Current biology*, 17(6), 514–519.

for Intelligent Physical Agents, F. (2002). Fipa communicative act library specification. `http://www.fipa.org/specs/fipa00037/`. Accessed: Aug. 2016.

Franklin, S. & Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent agents III agent theories, architectures, and languages* (pp. 21–35). Springer.

Froese, T. (2003). Steps toward the evolution of communication in a multi-agent system. In *Symposium for Cybernetics Annual Research Projects, SCARP*, volume 3 Reading, England.

Furlani, T. R., Jones, M. D., Gallo, S. M., DeLeon, R. L., White, J. P., Palmer, J. T., Simakov, N., Yearke, T., Sperhac, J., Rathsam, R., Laszewski, G. v., & Wang, F. (n.d). Acf xd metrics on demand (xdmod). `http://xdmod.acf.ku.edu`. Accessed: Aug. 2016.

Galantucci, B. & Steels, L. (2008). The emergence of embodied communication in artificial agents and humans. *Embodied communication in humans and machines*, (pp. 229).

Gerard, S. N. & Singh, M. P. (2013). Evolving protocols and agents in multiagent systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (pp. 997–1004). Saint Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems.

Ghavamzadeh, M. & Mahadevan, S. (2004). Learning to communicate and act using hierarchical reinforcement learning. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3 (pp. 1114–1121). New York, NY, USA: IEEE Computer Society.

Giles, C. L. & Jim, K.-C. (2002). Learning communication for multi-agent systems. In *Innovative Concepts for Agent-Based Systems* (pp. 377–390). Springer.

Gloor, P. A., Oster, D., Putzke, J., Fischbach, K., Schoder, D., Ara, K., Kim, T., Laubacher, R., Mohan, A., Olguin Olguin, D., Pentland, A., & Waber, B. N. (2007). Studying microscopic peer-to-peer communication patterns. In *American Conference on Information Systems* Keystone.

Gmytrasiewicz, P. J., Summers, M., & Gopal, D. (2002). Toward automated evolution of agent communication languages. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 10–19). Washington, DC, USA: IEEE.

Goldberg, D. E. & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1, 69–93.

Goldberg, D. E., Deb, K., & Clark, J. H. (1991). Genetic algorithms, noise, and the sizing of populations. *Urbana*, 51, 61801.

Goldman, C. V., Allen, M., & Zilberstein, S. (2004). Decentralized language learning through acting. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3 (pp. 1006–1013). New York, NY, USA: IEEE Computer Society.

Goldman, C. V. & Zilberstein, S. (2003). Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 137–144). Melbourne, VIC, Australia: ACM.

Goldreich, O., Juba, B., & Sudan, M. (2012). A theory of goal-oriented communication. *Journal of the ACM (JACM)*, 59(2), 8.

Goodman, P. S. (1986). Impact of task and technology on group performance. *Designing effective workgroup*, (pp. 120–167).

Guetzkow, H. & Simon, H. A. (1955). The impact of certain communication nets upon organization and performance in task-oriented groups. *Management science*, 1(3-4), 233–250.

Gupta, D. & Ghafir, S. (2012). An overview of methods maintaining diversity in genetic algorithms. *International Journal of Emerging Technology and Advanced Engineering*, 2(5), 56–60.

Gutiérez, C., García-Magariño, I., & Gómez-Sanz, J. J. (2009). Evaluation of multi-agent system communication in ingenias. In *Bio-Inspired Systems: Computational and Ambient Intelligence* (pp. 619–626). Springer.

Hansen, N. & Kern, S. (2004). Evaluating the cma evolution strategy on multimodal test functions. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature* (pp. 282–291). Birmingham, UK: Springer.

Harik, G. R. & Lobo, F. G. (1999). A parameter-less genetic algorithm. In *Genetic and Evolutionary Computation Conference*, volume 99 (pp. 258–267). Orlando, Florida, USA.

Haupt, R. L. (2000). Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors. In *2000 IEEE International Antennas and Propagation Symposium and USNC/URSI National Radio Science Meeting*, volume 2 (pp. 1034–1037). Salt Lake City, Utah, USA: IEEE.

Hedar, A.-R., Ong, B. T., & Fukushima, M. (2007). *Genetic algorithms with automatic accelerated termination*. Technical report, Department of Applied Mathematics and Physics, Kyoto University.

Heise, G. A. & Miller, G. A. (1951). Problem solving by small groups using various communication nets. *The Journal of Abnormal and Social Psychology*, 46(3), 327.

Herold, D. M. (1978). Improving the performance effectiveness through task-contingency selection of intervention strategies. *Academy of Management Review*, 3, 315–325.

Hirokawa, R. Y. (1980). A comparative analysis of communication patterns within effective and ineffective decision-making groups. *Communications Monographs*, 47(4), 312–321.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, MI, USA: U Michigan Press.

Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1), 22–26.

Hurt, D. (2005). An empirical evaluation of communication and coordination effectiveness in autonomous reactive multiagent systems. Master's thesis, University of North Texas.

Iba, H., Nozoe, T., & Ueda, K. (1997). Evolving communicating agents based on genetic programming. In *IEEE International Conference on Evolutionary Computation* (pp. 297–302). Indianapolis, IN, USA: IEEE.

Jim, K.-C. & Giles, C. L. (2000). Talking helps: Evolving communicating agents for the predator-prey pursuit problem. *Artificial Life*, 6(3), 237–254.

Juba, B. (2011). Theory of finite goal-oriented communication. In *Universal Semantic Communication* (pp. 27–58). Springer.

Juba, B. & Vempala, S. (2011). Semantic communication for simple goals is equivalent to on-line learning. In *The 22nd International Conference on Algorithmic Learning Theory* (pp. 277–291). Espoo, Finland: Springer.

Kazimipour, B., Li, X., & Qin, A. (2014). A review of population initialization techniques for evolutionary algorithms. In *2014 IEEE Congress on Evolutionary Computation* (pp. 2585–2592). Beijing, China: IEEE.

Khasteh, S. H., Shouraki, S. B., Halavati, R., & Khameneh, E. (2006a). Evolution of a communication protocol between a group of intelligent agents. In *2006 World Automation Congress* (pp. 1–6). Budapest, Hungary: IEEE.

Khasteh, S. H., Shouraki, S. B., Halavati, R., & Lesani, M. (2006b). Communication protocol evolution by natural selection. In *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* (pp. 152–152). Vienna, Austria: IEEE.

Kinney, M. & Tsatsoulis, C. (1998). Learning communication strategies in multiagent systems. *Applied intelligence*, 9(1), 71–91.

Kirchenbauer, L. R. (n.d.). Does leadership = communication?

use this communication matrix. http://www.exkalibur.com/nothing-but-leadership-does-leadership-communication/. Accessed: Aug. 2016.

Knaepkens, G., Bervoets, L., Verheyen, E., & Eens, M. (2004). Relationship between population size and genetic diversity in endangered populations of the european bullhead (cottus gobio): implications for conservation. *Biological Conservation*, 115(3), 403–410.

Koljonen, J. & Alander, J. T. (2006). Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms. In *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)* (pp. 54–60). Espoo, Finland: Citeseer.

Korejo, I., Yang, S., & Li, C. (2009). A comparative study of adaptive mutation operators for genetic algorithms. In *the VIII Metaheuristics International Conference* Hamburg, Germany.

Koumousis, V. K. & Katsaras, C. P. (2006). A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Transactions on Evolutionary Computation*, 10(1), 19–28.

Kumar, R., Narula, S., & Kumar, R. (2013). A population initialization method by memetic algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(4).

Kvasnicka, V. & Pospichal, J. (1999). An emergence of coordinated communication in populations of agents. *Artificial Life*, 5(4), 319–342.

Leavitt, H. J. (1951). Some effects of certain communication patterns on group performance. *The Journal of Abnormal and Social Psychology*, 46(1), 38.

Lee, C.-Y. & Yao, X. (2004). Evolutionary programming using mutations based on the lévy probability distribution. *IEEE Transactions on Evolutionary Computation*, 8(1), 1–13.

Leung, Y.-W. & Wang, Y. (2001). An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 5(1), 41–53.

Levin, M. (1995). The evolution of understanding: A genetic algorithm model of the evolution of communication. *BioSystems*, 36(3), 167–178.

Libelli, S. M. & Alba, P. (2000). Adaptive mutation in genetic algorithms. *Soft Computing*, 4(2), 76–80.

Lim, D., Ong, Y.-S., Jin, Y., & Sendhoff, B. (2006). Trusted evolutionary algorithm. In *2006 IEEE Congress on Evolutionary Computation* (pp. 149–156). Vancouver, BC, Canada: IEEE.

Lin, W.-Y., Lee, W.-Y., & Hong, T.-P. (2003). Adapting crossover and mutation rates in genetic algorithms. *Journal of Information Science and Engineering*, 19(5), 889–903.

Lincoln, J. R. & Miller, J. (1979). Work and friendship ties in organizations: A comparative analysis of relation networks. *Administrative Science Quarterly*, (pp. 181–199).

Lobo, F. G. & Goldberg, D. E. (2004). The parameter-less genetic algorithm in practice. *Information Sciences*, 167(1), 217–232.

Lobo, F. G. & Lima, C. F. (2005). A review of adaptive population sizing schemes in genetic algorithms. In *Proceedings of the 7th Annual Workshop on Genetic and Evolutionary Computation* (pp. 228–234). New York, NY, USA: ACM.

Lozano, M., Herrera, F., & Cano, J. R. (2005). Replacement strategies to maintain useful diversity in steady-state genetic algorithms. In *Soft Computing: Methodologies and Applications* (pp. 85–96). Springer.

Luke, S. (2013). *Essentials of metaheuristics.* Accessed: https://cs.gmu.edu/ sean/book/metaheuristics/Essentials.pdf.

Maaranen, H., Miettinen, K., & Penttinen, A. (2007). On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, 37(3), 405–436.

Mackin, K. J. & Tazaki, E. (2000). Unsupervised training of multiobjective agent communication using genetic programming. In *Proceedings of the Fourth International Conference on*

Knowledge-Based Intelligent Engineering Systems and Allied Technologies, volume 2 (pp. 738–741). Brighton, UK: IEEE.

Mackin, K. J. & Tazaki, E. (2002). Multiagent communication combining genetic programming and pheromone communication. *Kybernetes*, 31(6), 827–843.

Maclennan, B. (1990). *Evolution of Communication in a Population of Simple Machines*. Technical report, University of Tennessee, Knoxville, Department of Computer Science.

Magalhaes-Mendes, J. (2013). A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. *WSEAS Transactions on Computers*, 12(4), 164–173.

Mammen, D. & Lesser, V. (1997). A testbed for the evaluation of multi-agent communication and problem-solving strategies. In *1997 AAAI Workshop on Constraints and Agents* (pp. 32–38). Providence, Rhode Island, USA.

Marocco, D., Cangelosi, A., & Nolfi, S. (2003). The emergence of communication in evolutionary robots. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1811), 2397–2421.

Marocco, D. & Nolfi, S. (2005). *Emergence of Communication in Embodied Agents Evolved for the Ability to Solve a Collective Navigation Problem*. Technical report, Institute of Cognitive Science and Technology, CNR.

Marocco, D. & Nolfi, S. (2006a). Origins of communication in evolving robots. In *From Animals to Animats 9* (pp. 789–803). Springer.

Marocco, D. & Nolfi, S. (2006b). Self-organization of communication in evolving robots. In *Proceedings of the Conference on Artificial Life (ALIFE)* (pp. 178–184). Bloomington, IN.

Mataric, M. (1995). *Communication strategies for cooperating behaviour-based robots*. Technical report, AAAI.

McGrath, J. E. (1984). *Groups: Interaction and performance*, volume 14. Englewood Cliffs, NJ: Prentice-Hall.

Melo, F. S. & Spaan, M. T. (2011). A pomdp-based model for optimizing communication in multiagent systems. In *Proceedings of the 1st European Workshop on Multiagent Systems* Oxford, UK.

Meng, Y. & Jin, Y. (2012). Distributed multi-agent systems for a collective construction task based on virtual swarm intelligence. *Innovations and Developments of Swarm Intelligence Applications*, (pp. 308).

Miller, B. L. & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3), 193–212.

Mostafa, H. & Lesser, V. (2009). Offline planning for communication by exploiting structured interactions in decentralized mdps. In *2009 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, volume 2 (pp. 193–200). Milan, Italy: IET.

Mullen, B., Johnson, C., & Salas, E. (1991). Effects of communication network structure: Components of positional centrality. *Social Networks*, 13(2), 169–185.

Mühlenbein, H. & Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm i. continuous parameter optimization. *Evolutionary Computation*, 1(1), 25–49.

Nair, R., Roth, M., & Yohoo, M. (2004). Communication for improving policy computation in distributed pomdps. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 1098–1105). Washington, DC, USA: IEEE Computer Society.

Noever, D. & Baskaran, S. (1992). Steady state vs. generational genetic algorithms: A comparison of time complexity and convergence properties. *Preprint series*, (pp. 92–07).

Nolfi, S. (2005). Emergence of communication in embodied agents: Co-adapting communicative and non-communicative behaviours. *Connection Science*, 17(3-4), 231–248.

Nolfi, S. (2013). Emergence of communication and language in evolving robots. *New Perspectives on the Origins of Language*, 144, 533–554.

Nolfi, S. & Mirolli, M. (2009). *Evolution of communication and language in embodied agents*. Berlin, Germany: Springer Science & Business Media.

North, M., Howe, T. R., Collier, N. T., & Vos, J. (2005). The repast simphony runtime system. In *Proceedings of Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms* Argonne, Illinois, USA: Citeseer.

Ong, Y. S. & Keane, A. J. (2004). Meta-lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2), 99–110.

Ong, Y.-S., Lim, M.-H., Zhu, N., & Wong, K.-W. (2006). Classification of adaptive memetic algorithms: a comparative study. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(1), 141–152.

Parker, C. & Zhang, H. (2002). Robot collective construction by blind bulldozing. In *IEEE Conference on Systems, Cybernetics and Man* (pp. 177–195). Yasmine Hammamet, Tunisia.

Parker, C. A. & Zhang, H. (2006). Collective robotic site preparation. *Adaptive Behavior*, 14(1), 5–19.

Parker, C. A., Zhang, H., & Kube, C. R. (2003). Blind bulldozing: multiple robot nest construction. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2 (pp. 2010–2015). Las Vegas, NV, USA: IEEE.

Patel, R. & Raghuwanshi, M. (2010). Multi-objective optimization using multi parent crossover operators. *Journal of Emerging Trends in Computing and Information Sciences*, 2(2).

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000). Bayesian optimization algorithm, population sizing, and time to convergence. In *Genetic and Evolutionary Computation Conference* (pp. 275–282). Las Vegas, Nevada, USA.

Petersen, K., Nagpal, R., & Werfel, J. (2011). Termes: An autonomous robotic system for three-dimensional collective construction. In *Proceedings of Robotics: Science & Systems VII* Los Angeles, CA, USA.

Piszcz, A. & Soule, T. (2006). Genetic programming: Optimal population sizes for varying complexity problems. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (pp. 953–954). New York, NY, USA: ACM.

Potter, M. A. (1992). A genetic cascade-correlation learning algorithm. In *International Workshop on Combinations of Genetic Algorithms and Neural Networks* (pp. 123–133). Baltimore, Maryland, USA: IEEE.

Preist, C. & Pearson, S. (1998). An adaptive choice of messaging protocol in multi-agent systems. In *Proceedings of the International Conference on Multi Agent Systems* (pp. 457–458). Paris, France: IEEE.

Quinn, M. (2001). Evolving communication without dedicated communication channels. In *Advances in Artificial Life* (pp. 357–366). Springer.

Rawal, A., Rajagopalan, P., Miikkulainen, R., & Holekamp, K. (2012). Evolution of a communication code in cooperative tasks. In *International Conference on the Simulation and Synthesis of Living Systems*, volume 13 (pp. 243–250). East Lansing, Michigan, USA.

Reagans, R. & Zuckerman, E. W. (2001). Networks, diversity, and productivity: The social capital of corporate r&d teams. *Organization Science*, 12(4), 502–517.

Reddy, T. S., Zaruba, G., & Huber, M. (2012). A game theoretic framework for communication in fully observable multiagent systems. In *11th International Conference on Machine Learning and Applications*, volume 1 (pp. 697–702). Boca Raton, Florida, USA: IEEE.

Reeves, C. R. (1993). Using genetic algorithms with small populations. In *International Conference on Genetic Algorithms*, volume 590 (pp. 90–92). Urbana-Champaign, IL, USA.

Roeva, O., Fidanova, S., & Paprzycki, M. (2013). Influence of the population size on the genetic algorithm performance in case of cultivation process modelling. In *2013 Federated Conference on Computer Science and Information Systems* (pp. 371–376). Krakow, Poland: IEEE.

Rosenthal, E. (1997). Social networks and team performance. *Team Performance Management: An International Journal*, 3(4), 288–294.

Roth, M. (2007). *Execution-time communication decisions for coordination of multi-agent teams*. PhD thesis, Carnegie Mellon University.

Roth, M., Simmons, R., & Veloso, M. (2005). Reasoning about joint beliefs for execution-time communication decisions. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 786–793). New York, NY, USA: ACM.

Roth, M., Simmons, R., & Veloso, M. (2006). What to communicate? execution-time decision in multi-agent pomdps. In *Distributed Autonomous Robotic Systems 7* (pp. 177–186). Springer.

Russell, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, 94(1), 57–77.

Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education, 2 edition.

Rybski, P. E., Larson, A., Veeraraghavan, H., LaPoint, M., & Gini, M. (2007). *Distributed Autonomous Robotic Systems 6*, chapter Communication Strategies in Multi-robot Search and Retrieval: Experiences with MinDART, (pp. 317–326). Springer Japan: Tokyo.

Rylander, B. & Gotshall, S. (2002). Optimal population size and the genetic algorithm. *Population*, 100(400), 900.

Safe, M., Carballido, J., Ponzoni, I., & Brignole, N. (2004). *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings*, chapter On Stopping Criteria for Genetic Algorithms, (pp. 405–413). Springer Berlin Heidelberg: Berlin, Heidelberg.

Schaffer, J. D. & Morishima, A. (1987). An adaptive crossover distribution mechanism for genetic algorithms. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application* (pp. 36–40). Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Schuil, C., Valente, M., Werfel, J., & Nagpal, R. (2006). Collective construction using lego robots. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference* (pp. 1976–1977). Boston, MA, USA.

Searle, J. R. (1976). A classification of illocutionary acts. *Language in Society*, 5(01), 1–23.

Shaw, M. E. (1973). Scaling group tasks: A method for dimensional analysis. *JSAS Catalog of Selected Documents in Psychology*, 3(8).

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1), 51–92.

Singh, M. P. (2003). *Agent Communication Languages: Rethinking the Principles*, chapter Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies, (pp. 37–50). Springer Berlin Heidelberg: Berlin, Heidelberg.

Smith, R. E. & Smuda, E. (1995). Adaptively resizing populations: Algorithm, analysis, and first results. *Complex Systems*, 9(1), 47–72.

Sparrowe, R. T., Liden, R. C., Wayne, S. J., & Kraimer, M. L. (2001). Social networks and the performance of individuals and groups. *Academy of Management Journal*, 44(2), 316–325.

Speranzon, A. & Johansson, K. H. (2003). Distributed pursuit-evasion game: Evaluation of some communication schemes. In *Symposium on Autonomous Intelligent Networks and Systems* Menlo Park, CA.

Srinivas, M. & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4), 656–667.

Steels, L. (2003a). The evolution of communication systems by adaptive agents. In *Adaptive Agents and Multi-Agent Systems* (pp. 125–140). Springer.

Steels, L. (2003b). Evolving grounded communication for robots. *Trends in Cognitive Sciences*, 7(7), 308–312.

Steels, L. (2015). *The Talking Heads experiment: Origins of words and meanings*, volume 1. Berlin, Germany: Language Science Press.

Steels, L. & Bleys, J. (2005). Planning what to say: Second order semantics for fluid construction grammars. In *Proceedings of the Conference of the Spanish Association for Artificial Intelligence*, volume 5 galicia, spain.

Steiner, I. D. (1972). *Group Process and Productivity (Social Psychological Monograph)*. Richmond, TX, USA: Academic Press Inc.

Stewart, G. L. & Barrick, M. R. (2000). Team structure and performance: Assessing the mediating role of intrateam process and the moderating role of task type. *Academy of Management Journal*, 43(2), 135–148.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms* San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc.

Syswerda, G. (1991). A study of reproduction in generational and steady state genetic algorithms. *Foundations of Genetic Algorithms*, 2, 94–101.

Szer, D. & Charpillet, F. (2004). Improving coordination with communication in multi-agent reinforcement learning. In *16th IEEE International Conference on Tools with Artificial Intelligence* (pp. 436–440). Boca Raton, FL, USA: IEEE.

Thierens, D. (1997). Selection schemes, elitist recombination, and selection intensity. In *Proceedings of the 7th International Conference on Genetic Algorithms* (pp. 152–159). San Francisco, CA, USA.

Thierens, D. (2002). Adaptive mutation rate control schemes in genetic algorithms. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 1 (pp. 980–985). Washington, DC, USA: IEEE.

Tian, L., Luo, J., & Huang, Z. (2013). Communication based on interactive dynamic influence diagrams in cooperative multi-agent systems. In *8th International Conference on Computer Science & Education* (pp. 56–61). Colombo, Sri Lanka: IEEE.

Trianni, V., Labella, T. H., & Dorigo, M. (2004). Evolution of direct communication for a swarm-bot performing hole avoidance. In *Ant Colony Optimization and Swarm Intelligence* (pp. 130–141). Springer.

Tsai, J.-T., Liu, T.-K., & Chou, J.-H. (2004). Hybrid taguchi-genetic algorithm for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 8(4), 365–377.

Tu, Z. & Lu, Y. (2004). A robust stochastic genetic algorithm (stga) for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 8(5), 456–470.

Tushman, M. L. (1979). Work characteristics and subunit communication structure: A contingency analysis. *Administrative Science Quarterly*, (pp. 82–98).

Unhelkar, V. V. & Shah, J. A. (2016). Contact: Deciding to communicate during time-critical collaborative tasks in unknown, deterministic domains. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Van Oijen, J., Van Doesburg, W., & Dignum, F. (2011). Goal-based communication using bdi agents as virtual humans in training: An ontology driven dialogue system. *Agents for Games and Simulations II*, (pp. 38–52).

Vekaria, K. & Clack, C. (1998). Selective crossover in genetic algorithms: an empirical study. In *5th International Conference on Parallel Problem Solving from Nature—PPSN V* (pp. 438–447). Amsterdam, The Netherlands: Springer.

Wagner, K., Reggia, J. A., Uriagereka, J., & Wilkinson, G. S. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1), 37–69.

Wang, J. & Gasser, L. (2002). Mutual online concept learning for multiple agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 362–369). New York, NY, USA: ACM.

Wasserman, S. & Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge, United Kingdom: Cambridge university press.

Wawerla, J., Sukhatme, G. S., & Mataric, M. J. (2002). Collective construction with multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3 (pp. 2696–2701). lausanne, switzerland: IEEE.

Wei, C., Hindriks, K. V., & Jonker, C. M. (2014). The role of communication in coordination protocols for cooperative robot teams. In *International Conference on Agents and Artificial Intelligence. ICAART (2)* (pp. 28–39). Loire Valley, France.

Werfel, J. (2012). Collective construction with robot swarms. In *Morphogenetic Engineering* (pp. 115–140). Springer.

Werfel, J. & Nagpal, R. (2006). Extended stigmergy in collective construction. *IEEE Intelligent Systems*, 21(2), 20–28.

Werner, G. & Dyer, M. (1992). Evolution of communication in artificial organisms. In C. Langton, C. Taylor, D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 659–687). Redwood City, CA: Addison-Wesley Pub.

Werner, G. M. & M, D. (1991). Evolution of communication in artificial organisms. In *Proceedings of the Second International Conference of Artificial Life* (pp. 659–687). Redwood City, CA, USA.

White, T. & Oppacher, F. (1994). Adaptive crossover using automata. In *Parallel Problem Solving from Nature—PPSN III* (pp. 229–238). Springer.

Williamson, S., Gerding, E., & Jennings, N. (2008). A principled information valuation for communications during multi-agent coordination. In *AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains* (pp. 137–151).

Wischmann, S., Floreano, D., & Keller, L. (2012). Historical contingency affects signaling strategies and competitive abilities in evolving populations of simulated robots. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 109 (pp. 864–868).: National Academy Sciences.

Wooldridge, M. & Jennings, N. R. (1994). Agent theories, architectures, and languages: a survey. In *Intelligent Agents* (pp. 1–39). Springer.

Wu, A. S., Yu, H., Jin, S., Lin, K.-C., & Schiavone, G. (2004). An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Transactions on Parallel and Distributed Systems*, 15(9), 824–834.

Wu, F., Zilberstein, S., & Chen, X. (2009). Multi-agent online planning with communication. In *The International Conference on Automated Planning and Scheduling* Thessaloniki, Greece.

Wu, F., Zilberstein, S., & Chen, X. (2011). Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2), 487–511.

Xuan, P. & Lesser, V. (2002). Multi-agent policies: from centralized ones to decentralized ones. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 1098–1105). New York, NY, USA: ACM.

Xuan, P., Lesser, V., & Zilberstein, S. (2000). Formal modeling of communication decisions in cooperative multiagent systems. In *Workshop on Game-Theoretic and Decision-Theoretic Agents* Edmonton, Alberta, Canada: Citeseer.

Xuan, P., Lesser, V., & Zilberstein, S. (2001). Communication decisions in multi-agent cooperation: Model and experiments. In *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 616–623). New York, NY, USA: ACM.

Yahaya, B., Latip, R., Othman, M., & Abdullah, A. (2011). Dynamic load balancing policy with communication and computation elements in grid computing with multi-agent system integration. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1(3), 757–765.

Yao, X. (1993). An empirical study of genetic operators in genetic algorithms. *Microprocessing and Microprogramming*, 38(1), 707–714.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2), 82–102.

Yob, G. (1975). Creative computing. *The People's Computer Company*, 2, 51–54.

Yu, T.-L., Sastry, K., Goldberg, D. E., & Pelikan, M. (2006). *Population sizing for entropy-based model building in genetic algorithms*. Technical report, University of Illinois.

Zhang, G., Wang, S., & Li, Y. (2010). A self-adaptive genetic algorithm based on the principle of searching for things. *Journal of Computers*, 5(4), 646–653.

Zhang, Y. (2006). Observant and proactive communication in multi-agent teamwork. In *2006*

IEEE/WIC/ACM International Conference on Intelligent Agent Technology (pp. 460–466). Hong Kong: IEEE.

Zhang, Y. (2007). *Proactive communication in multi-agent teamwork*. PhD thesis, Texas A&M University.

Zhong, W., Liu, J., Xue, M., & Jiao, L. (2004). A multiagent genetic algorithm for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(2), 1128–1141.

Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J., & Lum, K. Y. (2007). Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(1), 66–76.