ABSTRACT

Title of Thesis:	A METHOD FOR IMPROVING DECENTRALIZED TASK ALLOCATION FOR MULTIAGENT SYSTEMS IN LOW-COMMUNICATION ENVIRONMENTS
	Oghenetekevwe E. Akoroda,
	Master of Science, 2021
Thesis Directed By:	Professor Jeffrey W. Herrmann, Department of Mechanical Engineering

Communication is an important aspect of task allocation, but it has a cost and low communication restricts the information exchange needed for task allocation. As a result, a lot of decentralized task allocation algorithms perform worse as communication worsens.

The contribution of this thesis is a method to improve the performance of a task allocation algorithm in low-communication environments and reduce the cost of communication by restricting communication. This method, applied to the Consensus Based Auction Algorithm (CBAA), determines when an agent should communicate and estimates the information that will be received from other agents.

This method is compared to other decentralized task allocation algorithms at different levels of communication in a ship protection scenario. Results show that this method when applied to CBAA performs comparably to CBAA while reducing communication.

A METHOD FOR IMPROVING DECENTRALIZED TASK ALLOCATION FOR MULTIAGENT SYSTEMS IN LOW-COMMUNICATION ENVIRONMENTS

by

Oghenetekevwe E. Akoroda

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2021

Advisory Committee:
Professor Jeffrey Herrmann, Chair and Advisor
Assistant Professor Michael Otte,
Professor Shapour Azarm

Dedication

I would like to dedicate this work to my team in the AFRL research group, whose research and advice has paved the way for the contributions of this paper and without which this work would not be possible.

Acknowledgements

I am grateful to all the people whose work, advice and support were critical in helping me complete this thesis.

First of all, I would like to thank my advisor, Prof. Jeffrey Herrmann for his decision to employ me as a graduate research assistant in his lab. His continual support and advice throughout my time as a graduate research assistant and the research I did in my first year inspired me to write this thesis. I also appreciate his patience and all the helpful suggestions he proffered as I wrote this paper and conducted this research.

I also want to thank the other professors in the AFRL Lab research group such as Dr. Michael Otte, Prof. Shapour Azarm and Dr. Huan Xu. It has been a pleasure working with you and learning from you. I am also thankful for all the advice you offered when I and my partner were working on the metareasoning survey paper.

I am also full of gratitude to my fellow students in the AFRL research group such as Samuel Langlois, Sharan Nayak, Estefany Carrillo, Suyash Yeotikar. The previous work you did in building the multiagent simulation to run comparison experiments for task allocation algorithms greatly reduced my workload and for that, I am grateful. I am also grateful for your helpful suggestions and for answering my questions on the simulation.

I am thankful to my family for their prayers and support over the years and also while I was working on my thesis. I am especially thankful to my brother for his help and suggestions and the intellectual discussions we had. Those source of ideas proved to be useful.

Lastly, I am grateful to AFRL Lab for the financial support they provided for me to conduct research in a fast growing field. This work was supported by the grant FA8750-18-2-0114 from the Air Force Research Laboratory (AFRL), USA.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

This chapter contains sections that introduce several concepts that relate to the topic of this thesis and provides information on the organization of this thesis. Section 1.1. discusses multiagent systems including the different types while Section 1.2. defines the concept of metareasoning and discusses some metareasoning problems and techniques. Section 1.3. provides a brief overview of task allocation; Section 1.4. introduces the research question and Section 1.5. provides an overview of the subsequent chapters in this thesis.

1.1 Multiagent Systems

Multiagent systems are systems consisting of multiple agents. They are being increasingly used in search and rescue [1], surveillance, [2] and firefighting [3], because they can accomplish complex missions quickly, efficiently and cheaply. Multiagent systems can be divided into collaborative and non-collaborative multiagent systems. In collaborative multiagent systems, the agents in the system coordinate with each other to achieve a goal. However in non-collaborative multiagent systems, the agents in the system do not collaborate with each other. In some cases, the agents may have competing objectives. Multiagent systems can be divided into heterogeneous and homogeneous multiagent systems. In homogeneous multiagent systems, the agents in the system are identical and can perform identical tasks. These systems are highly fungible, i.e., each agent can do the task of every other agent in an identical fashion. On the other hand, in heterogeneous multiagent systems, the agents in the system are not identical. They may have different properties such as speed and size; different capabilities such as sensors, which can affect their ability to perform different tasks. In some cases, heterogeneous systems can have specialized agents, which are capable of performing certain tasks. Thus, not all tasks can be performed by all agents and agents can perform differently on identical tasks.

1.2 Metareasoning

Metareasoning can be defined as reasoning about reasoning [4]. It is a higher level of reasoning that aims to improve an agent's performance by reasoning about and controlling the agent's decision making processes. For example, consider a robot using a motion planning algorithm to find a valid path from its current location to a destination. Metareasoning can help the agent determine the right time to stop computing and start executing a plan. It can also help the agent identify ways to improve the performance of the algorithm. In a metareasoning context, an agent can be described as having three levels, as shown in Figure 1.1.

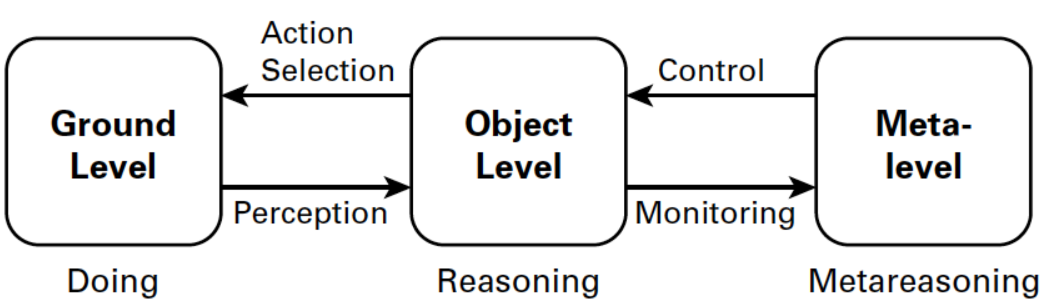

Figure 1.1: Duality in reasoning and acting [4]

The ground-level consists of actions that are taken directly by the agent to accomplish its goal. These actions influence the environment and can also change the state of the agent. In the example above, the movement of the robot along the path selected by the motion planning algorithm would be a ground-level action. Object-level actions are generally computations made by the agent to understand its environment and determine the configuration of ground-level actions to take that best achieves its goal. The computations performed by the motion planning algorithm would be the object-level action. Meta-level actions are higher-level computations that the agent performs to improve the performance of its object-level actions. A meta-level action would be a computation performed by the agent to select a motion planning algorithm.

Metareasoning can help agents adapt to dynamic and novel environments by improving the performance of different facets of its computations. Some metareasoning problems that have been studied in the literature include the algorithm selection problem [5], the deliberation problem [6], and the allocation problem [7]. The algorithm selection problem addresses the challenge of selecting the algorithm that performs best, out of a set of algorithms, for the given agent and scenario. The performance metric used to assess the algorithm will influence the choice of algorithm. The deliberation problem addresses sub-problems like should the agent think or act and should the agent gather more information or act that focus on the right time to transition from deliberation to execution. The allocation problem addresses sub-problems like allocating computational time to different problems and allocating time to different computations the agent has to perform that focus on optimization of the agent's resources.

Metareasoning is not only useful for individual agents, but also for multiagent systems (MAS) as MAS presents a whole new set of challenges. In a collaborative MAS where agents work together to achieve a goal, agents face additional challenges like effective collaboration. Agents in a MAS have to reason about the actions and capabilities of other agents with respect to the goal and the environment. In addition, the agents have to coordinate their behaviors and achieve consensus in order to fulfil their goals. This can introduce further complexity into the agent's computations. These present problems like the teaming problem [8], the task allocation problem [9], and the communication problem [10]. The teaming problem occurs in situations where agents team with a subset of other agents in order to achieve a goal. It addresses the problem of determining the best team for an agent. The task allocation problem, the focus of this paper, addresses the challenge of assigning tasks to the agents in the MAS. The communication problem addresses the challenge of sharing information with other agents when communication consumes resources.

Different metareasoning techniques have been developed and studied in literature for tackling the litany of metareasoning problems [11]. These techniques include redefining relationships in MAS, modifying parameters of an algorithm, and modifying reasoning rules.

1.3 Task Allocation

The task allocation problem is a problem that has been widely studied in literature. It can be described as follows: Given a certain number of agents and number of tasks, find the assignment of tasks to agents that conforms to the constraints of the system, e.g., some tasks can only be performed by certain agents, time limit on certain tasks and optimization of some metric, i.e., utility or cost.

The tasks that are being assigned can take on different forms and have different constraints. These include tasks that require a single robot to be executed, tasks that require multiple robots to be executed and primitive tasks, i.e., tasks that cannot be further simplified [12]. Tasks can also have constraints such as the following:

- Partial ordering i.e. which task must be completed before or after a set of others (but not necessarily immediately before or after),
- Time windows, in which a task must be completed in a given time frame, or before a certain deadline,
- Coupling, in which two or more tasks must be executed at the same time,
- Incompatibility, in which executing one task may preclude or obsolete the execution of others.

The system may also impose constraints on the task assignment problem such as the following:

- Mobility interferences, due to narrow spaces in relation to robot size or numbers.

- Network range, due to limited coverage of infrastructure, ad-hoc devices with limited range or a need for line-of-sight.

1.4 Research Question

The task allocation problem is a problem that has been widely studied in literature. I researched different approaches to solving the problem when there is low communication and when communication has a cost. A lot of these approaches are explained in the next chapter. The question I seek to answer in this thesis is: Is metareasoning a viable approach to improving performance of a multiagent system in low communication while reducing communication costs? The contribution of this work is a metareasoning method to improve the performance of a task allocation algorithm in low-communication environments and reduce the cost of communication by restricting communication.

1.5 Thesis Outline

Chapter 2 of this thesis discusses work that has been done to solve the task allocation problem in the literature. It also discusses different approaches that have been taken to solve this problem for MAS in low or limited communication environments.

Chapter 3 defines relevant terms, states the assumptions and defines the problem that is at the core of this thesis. It also discusses the communication model that was used for running experiments.

Chapter 4 describes and analyzes the proposed task allocation algorithm. Chapter 5 describes 5 existing task allocation algorithms (CBAA, ACBBA, PIA, DHBA, HIPC) that have been compared with the new method applied to CBAA.

Chapter 6 describes the experimental setup that was used to run simulations to compare the new method applied to CBAA with the existing task allocation algorithms. It provides a description of the simulation framework and the design of experiments.

Chapter 7 provides the results of the comparison experiments that were run, and Chapter 8 discusses and analyzes the results including the limitations of these algorithm. Finally, Chapter 9 addresses conclusions based on the results of these paper and proposes future work of study.

Chapter 2: Literature Review

This chapter describes the work that has been done in reviewing the literature. It starts off with describing different task allocation algorithms and approaches that have been used to solve the task allocation problem. Afterwards, it delves into research that has been done on the impact of communication on task allocation and explains some approaches that have been used to resolve this issue. Finally, it ends with a discussion of the contributions of this thesis.

The task allocation problem has been widely studied in the literature and there have been numerous algorithms that have been proposed to solve this problem in many of its different contexts and forms as described in the previous chapter.

One of the more common approaches to solving this problem is the auction approach. In this approach, agents in a MAS place bids on different tasks and the agent with the highest bid is awarded the task. This approach can be applied in a centralized fashion with a single agent acting as the auctioneer and awarding the tasks or in a decentralized fashion with agents receiving bid information from one another and determining the winning bids and agents for each task. Decentralized task allocation algorithms like the Consensus-Based Auction Algorithm (CBAA), the Consensus Based Bundle Algorithm (CBBA) [13], the Performance Impact Algorithm (PIA) [14], the Hybrid Information and Plan Consensus Algorithm (HIPC) [15] are based on this approach.

Another approach to solving the task allocation problem is the use of optimization techniques which can be deterministic or stochastic. The Decentralized Hungarian-Based Algorithm (DHBA) [16] uses deterministic optimization techniques via the Hungarian method. On the other hand, the ant-colony optimization algorithm [17] and the decentralized genetic algorithm (GA) [18] employ stochastic optimization techniques for task allocation.

A different approach employs metareasoning to select the algorithm, out of a set of algorithms, that is expected to perform best. Herrmann [19] employed data analysis and machine learning to predict the performance of different algorithms and then tested a metareasoning approach that selects the algorithm with the best predicted performance.

Decentralized task allocation algorithms depend on communication between agents in order to assign tasks to each agent. As a result, different studies have been conducted to observe how a change in communication quality affects the performance of different task allocation algorithms. Nayak *et al.* [20] compared the performance of 5 decentralized task allocation algorithms at different communication conditions in different scenarios using 3 different communication models. Their results show that the performance of different task allocation algorithms deteriorates at worse communication conditions and different algorithms perform better than others either at low or high communication. Otte *et al.* [21] compared different auction algorithms and found out that their performance degrades in different ways as communication quality decreases. These studies show that algorithms that work well at perfect communication can perform poorly at low communication. As a result, work has been done to design algorithms for low communication conditions. Sujit *et al.* [22] proposed a team theory approach for task allocation under no communication and agents with limited sensor range. Dai *et al.* [23] presented an algorithm based on the principle of incomplete information game theory to solve the problem of task allocation under unreliable communication. Cheng et al. [24] presented an algorithm that employs a local sensing control law and time synchronization to distribute UAVs, with GPS and synchronized clocks but no communication, along a common curve such that consensus is achieved. Carrillo et al. [25] developed a metareasoning policy that switches the task allocation algorithm used by an individual agent as a function of the observed level of communication.

Previous research has investigated the use of approaches like game theory [23] , team theory [22] and a switching policy [25] to enable agent to make the best decisions or select the best task allocation algorithms under low communication. Other approaches have used sensor information as a form of implicit communication i.e. agents can detect other agents and use this to obtain information about the agent and the environment. However, these approaches do not typically focus on optimizing the communication and do not consider the cost of communication.

The main idea of this work is to use estimation as a substitute for communication so that the cost of communication can be reduced and the performance of a task allocation algorithm in low communication can be improved. Consequently, a valuation function is developed to assesses the marginal utility of sending a message. Messages are sent when the marginal utility is above a defined threshold. In addition, each agent continuously estimates the states of other agents and uses this information in the absence of communication. The agent uses communication from other agents to improve its estimate and the cycle continues. This method is applied to an existing algorithm (the base algorithm) and its performance is compared with the base algorithm and other algorithms.

Chapter 3: Preliminaries

In this chapter, the scenario studied in this thesis is described including any assumptions that were made. It also includes a definition of the task allocation problem, including a general case and the case relevant to this thesis. Lastly, it includes a discussion of the communication model that was used.

3.1 Assumptions

In the scenario considered in this thesis, given a set of autonomous agents and a set of tasks, each agent can be assigned to only one task, every task requires only one agent to be completed and the tasks can be performed concurrently. The scenario being considered is a ship protection scenario where agents have to search for and intercept targets to prevent them from hitting a ship moving through the workspace. Targets in the workspace can be classified as adversarial or non-adversarial targets. Adversarial targets are targets which have been programmed to hit the ship while non-adversarial targets are targets that move about randomly in the workspace and have not been programmed to hit the ship. Adversarial targets are considered neutralized when an agent tracks and then visits them. Targets are considered visited when an agent comes within a threshold distance from the target. The location of the targets is unknown to the agents; thus agents have to search for targets, and they have been equipped with sensors to detect and classify targets in the workspace. Although computational effort of an algorithm is important, in these experiments, it has a minimal impact on the performance of the agents and is thus not considered. The metrics used to compare the algorithms are

1. the number of hits to the ship
2. the maximum distance needed to travel in order to track adversarial targets.

The total number of messages sent by agents when utilizing each algorithm is also compared but is not defined as a formal metric. I have made the following assumptions about the problem:

1 Every task can be performed by each agent and all tasks are compatible with each other.

2 There are no collisions between agents or any other mobility considerations like robot size or restricted regions in the workspace.

3 Agents have equal capabilities to perform a task and do not have any energy consumption constraints.

4 There are no hard constraints on the timing and order of task completion.

5 Agents do not experience failure of any kind and are able to complete the tasks assigned to them

3.2. Problem Definition

The general task allocation problem is defined as follows:

Given a set of agents $A = \{a_1, \dots, a_n\}$ and a set of tasks $T = \{t_1, \dots, t_m\}$, find an assignment of tasks, Q_i for each agent such that $Q_1 \cup Q_2 \cup \dots \cup Q_n = T$ that is subject to the constraints of the system such as compatibility or coupling constraints.

In the ship protection scenario, a ship moves through a workspace that contains different targets such as adversarial targets, which are targets that try to move to the ship's location and hit the ship. Other targets in the workspace are classified as non-adversarial targets as they don't try to hit the ship. Agents search for these adversarial targets and track them, so they do not hit the ship. The workspace is divided into different regions called cells that agents can search.

The task allocation problem for the ship scenario is thus defined as follows:

Given a ship moving through a workspace, a set of agents $A = \{a_1, \dots, a_n\}$, a set of targets $T = \{ t_1, \dots, t_m\}$ with a subset of adversarial targets $AD = \{ad_1, \dots, ad_n\}$ with target locations unknown to agents, determine an assignment of tasks to agent that locates adversarial targets, minimizes the number of hits to the ship and minimizes the maximum distance it takes an agent to track adversarial targets.

3.3 Communication Model

The communication model being used to run experiments is the Rayleigh-Fading model [26] which attempts to model real world communication between agents. The Rayleigh-Fading model is a model of the propagation effects on a wireless signal. It is based on the assumption that the magnitude of a signal will vary randomly according to a Rayleigh distribution. In this model, there are two main parameters that affect signal strength: fading and path loss. Fading is the attenuation of the signal power due to objects, such as buildings in the environment that scatter the signal before it reaches the receiver. Interference from objects in the environment cause the signal to propagate along multiple paths that experience different shifts in amplitude, frequency and phase; thus, resulting in a constructive or destructive interference of the signal.

Path loss is the attenuation of the signal power due to increasing distance between the transmitter and the receiver. It is modeled by the equation below:

$$P_{PL} = P_{L_0} + 10\,\gamma log_{10}\frac{d}{d_0}$$

where P_{PL} = path loss

P_{L_0} = path loss at a reference distance, d_0

γ = path loss exponent

d = distance between transmitter and receiver

d_0 = reference distance

Figure 3.1 shows the attenuation of signal power due to the path loss and fading components

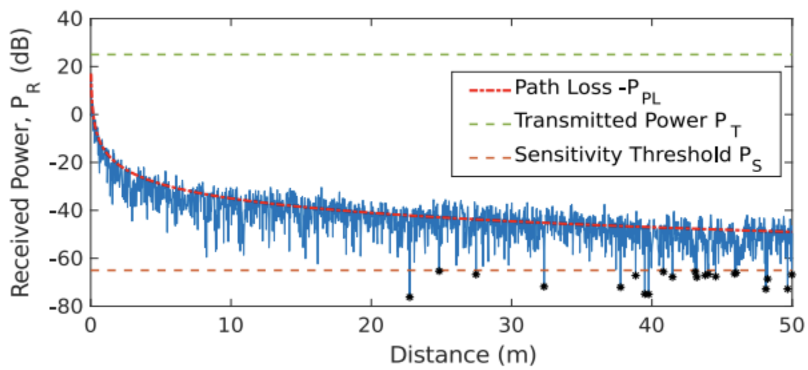

Figure 3.1: Attenuation of signal strength due to fading and path loss [20]

The total power loss is a contribution of the fading power, P_F and the path loss, P_{PL}. The transmitted power, P_T and the received power, P_R are related by the below equation.

$P_R = P_T - P_{PL} - P_F$

In experiments, a sensitivity threshold, P_S is defined such that the signal is received if the received power is greater than the sensitivity threshold.

Chapter 4: The Method Description

In this chapter, I propose a method that can be used to improve the performance of a task allocation algorithm in low communication environments. The principle of the method is described in Section 4.1 while its implementation in a specific task allocation algorithm , the Consensus-Based Auction Algorithm (CBAA) is described in Section 4.2. Although the CBAA algorithm has been described in Chapter 5, it will be described again for the purpose of explaining the implementation of this method. The goal of this method is to optimize the communication between agents in a multiagent system.

4.1 Method Motivation and Principles

The method consists of two sub-methods which are listed below:

1. Sub-method 1 determines when the agents should communicate
2. Sub-method 2 estimates the information of other agents.

<u>Sub-Method 1</u> (When should agents communicate)

Motivation - Frequent communication helps agent in a multiagent system to achieve information consistency quickly. However, communication has a cost which can negate the advantage brought about by frequent communication.

Principle - A function for computing the value of communication is developed. Value of communication is defined as the marginal utility of sending a message to other agents as measured by the last message sent. An agent runs the function to compute the value of communication and sends a message only if the value of communication exceeds a threshold value.

Sub-Method 2 (estimate information from other agents)

Motivation - Different task allocation algorithms require agents to exchange different kinds of information that each agent can use to compute the allocation of tasks. In auction algorithms like CBAA, CBBA, the agents exchange information on their winning bids list, which contains the agent's most up-to-date estimate of the highest bid made for each task thus far. In a multi-assignment algorithm like CBBA, the agents also exchange a winning agents list. In an optimization algorithm like the Decentralized Hungarian algorithm, the agents exchange information on their estimate of the global cost matrix. In a low communication environment, information from other agents is not readily available and this lack of information can impair the task allocation process

Principle – An agent uses its knowledge base, including information previously received from other agents, to estimate the information of other agents. When the information can be expressed mathematically as a number, the agent computes a range for the mathematical value. For example, in an auction algorithm, an agent will compute a range for the bids other agents place on different tasks.

This method is different from CBAA in the following ways:

1 In CBAA, each agent calculates its own bid on each incomplete task while in this method, agents also estimate the bids of other agents on each incomplete task

2 In CBAA, agent sends message at the end of every iteration; in this method, agents only send messages when the current value of communication exceeds a message threshold.

3 In CBAA, an agent determines task assignments by comparing its bids to bids in its winning bids list while in this method, task assignments are determined by comparing the bids to bids in the winning bids list and an agent's estimate of the best bids of other agents.

4.2 Method Implementation in a Task Allocation Algorithm

This section describes the details of implementing the above two sub-methods in a task allocation algorithm. The chosen algorithm is the Consensus-Based Auction Algorithm (CBAA). CBAA consists of two phases: the auction phase and the consensus phase

Auction Phase

In this phase, agents place bids on tasks asynchronously (at different times). Bids are ordered from low to high with the best bid having the lowest value. An agent stores a winning bids list which contains what the agent believes are the current best bids for different tasks. These best bids can be bids placed by the agent or placed by other agents. Each agent maintains a task list, which contains a list of the tasks and the current tasks assigned to the agent . An agent places bids on all the tasks and compares these bids to the bids in its winning bids list. The agent determines the available tasks as tasks for which it has a better bid than its winning bid estimate. The agent then assigns itself the task which has the best bid out of all its available tasks

Consensus Phase

In this phase, agents use the process of consensus to converge on the list of winning bids and determine the task assignment. Agents exchange their winning bids list with all other agents for which there exists a communication link. Each agent updates its winning bids list with the winning bids list received from other agents. An agent i performs this update process as follows:

Given a set of agents $A = \{a_1, a_2, \ldots, a_i, \ldots, a_m\}$, a set of tasks $T = \{t_1, t_2, \ldots, t_j, \ldots, t_n\}$ and a winning bids list $Y_i = \{y_{i1}, y_{i2}, \ldots, y_{ij}, \ldots, y_{in}\}$ for agent i

where y_{ij} is agent i's estimate of the winning bid for task j

$$y_{ij} = max_k\, y_{kj}$$

where $a_k \in A$

The agent unassigns itself from a task for which another agent has a better bid.

An iteration is a single run of the auction phase and the consensus phase. In this implementation of the algorithm, every agent has the same iteration count.

Bid

In the ship protection scenario, there are two types of tasks. The first task is visiting cells in the workspace to search for new moving targets while the second task is tracking these moving targets. The functions used to compute the bids the agents place on each task is described below.

If the task is searching cells

$$Bid = k_1 P^{tlv} \times DTS + (k_0 + 0.1 \times random) \times DTA$$

If the task is tracking moving targets

$$Bid = \frac{-k_2}{DTS + random} + (k_0 + 0.1 * random) * DTA$$

where DTS = distance from target to ship

DTA = distance from target to agent

tlv = the elapsed time since that cell was last visited by any agent

random = random number in [0 1)

P = probability term with a value of 0.9

k_0 is a function weight that places priority on the distance of an agent from a target.

k_1 is a function weight that places priority on visiting cells that have not been visited by an agent in a long time and that are close to the ship.

k_2 is a function weight that places priority on visiting moving targets that are closer to the ship.

The cost function for searching cells drives agents search cells that have not been visited in a long time and that are close to the ship. It also drives agents that are closest to a cell to search that cell. The cost function for tracking targets drives agents that are closest to a target to search that target and drives agents to track targets close to the ship. These cost functions and their corresponding function weights were adapted from the work of Estefany et al. [25]

Sub-Method 1

The value of communication (VOC) measures the marginal utility of new information obtained since the last message sent to other agents. Each agent estimates the bids of other agents and the marginal utility is the uncertainty in the estimate of the bids of the agent. The value of communication uses the cost function and the information obtained since the previous message sent to compute this marginal utility. The function used to calculate the value of communication is described below:

$$DTSmax = maximum\big(DTS(i-1), DTS(i)\big)$$

$$DTSmin = minimum\big(DTS(i-1), DTS(i)\big)$$

$$a1 = k_1 P^{tlv} * DTSmax - k_1 P^{tlv} * DTSmin$$

$$a0 = (k_0 + 0.1) * DTAmax - (k_0) * DTAmin$$

$$a2 = \frac{k_2}{DTSmax + 1} - \frac{k_2}{DTSmin}$$

where DTS(i-1) = distance of target from ship in the previous message sent

DTS(i) = current distance of target from ship

DTA(i-1) = distance of target from agent in the previous message sent

DTA(i) = current distance of target from agent

If the task is searching cells

$$VOC = a0 + a1$$

If the task is tracking moving targets

$$VOC = a0 - a2$$

Agents exchange information on winning bids list, list of targets and latest visit times for cells in the workspace with each other. An agent sends a message if there have been any changes in this information. If there has been no change in this information, an agent computes the VOC of sending a message at the current time and if the value of the VOC is greater than a threshold value, the message is sent. This threshold value represents the maximum uncertainty in the agent's winning bids that is allowed. If the value was too small, it would hardly reduce the number of messages being sent and if it was too large, there would be a significant delay in information as the environment changes and messages get dropped. A value of 10 was chosen by trial and error as it was small enough to ensure a small delay in information and large enough to decrease the number of messages sent compared to a value of 0.

<u>Sub-Method 2</u>

In this method, an agent i estimates the range of local bids of other agents based on its current information. Agent i first determines its latest location update on each agent in its list of agents and the timestamp at which that information was received. It uses that information to estimate the maximum and minimum possible bid for each agent. The inputs to this method are the (1) distance of each incomplete target in an agent's target list to the ship, (2) the last locations of other agents and (3) the latest visit times for cells in the workspace. An agent uses the equations below to compute the minimum and maximum bids for every other agent on each incomplete task. Incomplete tasks include all cells in the workspace and untracked moving targets.

$$a1 = k_1 P^{tlv} * DTS$$

$$a0min = (k_0) * DTA_{min}$$

$$a0max = (k_0 + 0.1) * DTA_{max}$$

$$a2min = \frac{k_2}{DTS + 1}$$

$$a2max = \frac{k_2}{DTS}$$

$$minimum\ Bid\ (moving\ targets) = a0\ min + a1$$

$$maximum\ Bid\ (moving\ targets) = a0\ max + a1$$

$$minimum\ Bid\ (cell\ targets) = a0\ min - a2max$$

$$maximum\ Bid\ (cell\ targets) = a0\ max - a2min$$

where DTA_{max} = maximum possible distance of target from agent

DTA_{min} = minimum possible distance of target from agent

dt = current time – timestamp of latest location update for an agent

speed = speed of an agent (agents have the same speed)

DTA_{prev} = distance of current location of target from last location of an agent

The calculation of DTA_{min} and DTA_{max} is beyond the scope of this paper

After an agent computes the minimum and maximum bids for every other agent on each incomplete task, it computes a range for the best bids for each task below:

Let $LBE_{kj,min}$ and $LBE_{kj,max}$ be the computed minimum and maximum local bid of agent k on task j. Then $BB_{j,min}$ and $BB_{j,max}$ the minimum and maximum value of the best bid other agents can place on task j is computed as follow

$$BB_{j,min} = min_k\ LBE_{kj,min}\ , BB_{j,max} = min_k\ LBE_{kj,max}$$

The estimate of the best bids other agents can place on each task is discarded if the range of the estimate is greater than a threshold value of 60. A much larger value would been very similar to the reference case where there are no estimates because it would hardly impact an agent's task assignment. A much smaller value would have resulted in the estimates being discarded too frequently. 60 was chosen as a good number to balance these effects. An agent only assigns itself a task that has the smallest bid of all available tasks. Available tasks are tasks where the agent's local bid for that task is greater than the believed winning bid for that task and a value X where

X = minimum winning bid estimate

+ 0.25*(maximum winning bid estimate – minimum winning bid estimate)

Chapter 5: Existing Decentralized Task Allocation Algorithms

This chapter describes the five task allocation algorithms that have been compared with the method applied to CBAA. It includes a discussion of the different phases and inputs of each algorithm.

5.1 Consensus-Based Auction Algorithm (CBAA)

The Consensus-Based Auction Algorithm [13] is a single-task allocation algorithm based on the principle of auctions. This algorithm has two phases: an assignment phase and a consensus phase. In the assignment phase, agents place bids on all tasks believed to be incomplete and assign themselves the lowest bid task. The agents then update their winning bids list with the lowest bid task and sends that list to other agents. In the consensus phase, agents receive winning bids lists from other agents and update their bids list with the lowest bids. Tasks are assigned to the agent that has the lowest bid.

The algorithm has the following inputs: target list, current distance travelled by each agent and an iteration count (number of times the assignment phase and the consensus phase are run iteratively).

5.2 Asynchronous Consensus-Based Bundle Algorithm (ACBBA)

This algorithm [27] is an extension of the Consensus-Based Bundle Algorithm (CBBA) [13]. CBBA is a multi-task allocation algorithm that is based on the principle of auctions like CBAA and similarly has two phases: an assignment phase and a consensus phase. In the assignment phase, each agent constructs an ordered bundle of tasks by adding tasks in a greedy fashion. The size of the bundle is restricted by the limit on the number of tasks an agent can perform. Agents place bids on tasks in their bundles and maintain a winning bids list which is updated with the bids on the task list. In the consensus phase, the agents broadcast their winning bids list along with the winning time stamps to other agents. They also receive messages from other agents and update their internal bid list. The algorithm has decision rules which specify how an agent updates its bid list after receiving a message from another agent and guarantees conflict-free assignments. ACBBA extends CBBA by allowing agents to communicate asynchronously in the consensus phase.

The algorithm has the following inputs: target list, current distance travelled by each agent, bundle size (maximum number of tasks in the bundle) and an iteration count (number of times the assignment phase and the consensus phase are run iteratively).

5.3 Performance Impact Algorithm (PIA)

This algorithm [14] is a heuristic multi-task allocation algorithm that introduces a concept called significance which measures the contribution of a task to the local cost generated by the agent. The algorithm has two phases: a task inclusion phase and a task removal and consensus phase. In the task inclusion phase, the agent calculates the significance of all tasks not included in its bundle and uses this to update the task bundle and the significance list. This significance list is broadcasted to other agents. In the task removal and consensus phase, the agent receives significance lists from other agents and tries to achieve consensus on the significance value of each task. The agent updates its bundle by removing tasks for which another agent has a lower significance value.

The algorithm has the following inputs: target list, current distance travelled by each agent, bundle size (maximum number of tasks in the bundle) and an iteration count (number of times the assignment phase and the consensus phase are run iteratively).

5.4 Decentralized Hungarian-Based Algorithm (DHBA)

This algorithm [16] is a single-task allocation algorithm is based on the Hungarian algorithm, which generates an optimal solution for an assignment linear programming problem. The Hungarian algorithm is a centralized approach and DHBA extends it for decentralized applications. In DHBA, a cost matrix is initialized using the current distance traveled and the current cost of completing incomplete tasks. This algorithm has two phases: an assignment phase and an update phase. In the assignment phase, each agent runs the Hungarian algorithm on the cost matrix to get an incomplete task. In the update phase, each agent receives a cost matrix from other agents and updates its cost matrix with the costs of other agents.

The algorithm has the following inputs: target list, current distance travelled by each agent, and an iteration count (number of times the assignment phase and the consensus phase are run iteratively).

5.5 Hybrid Information and Plan Consensus Algorithm (HIPC)

This algorithm [15] is a multi-task allocation algorithm in which agent create and update a bid space for all tasks and agents. The algorithm is initialized with an initial bid space, an available task set, and a neighborhood set, set of agents that each agent has situational awareness over. It has two phases: a local bid space creation phase and a consensus phase. In the local bid space creation phase, each agent computes an updated local bid space from the current bid space and the neighborhood set. In the consensus phase, each agent shares its bundle with its neighbors and checks if convergence has been achieved in its bid space.

The algorithm has the following inputs: local bid space, neighborhood set, target list, current distance travelled by each agent, bundle size (maximum number of tasks in the bundle) and an iteration count (number of times the assignment phase and the consensus phase are run iteratively).

Chapter 6: Experimental Setup

This Chapter provides in Section 6.1., a high level description of the simulation used to run the experiments. Section 6.2. specifies the parameters of the experiments that were ran and provides a range for those parameters.

6.1 Experimental Framework

In Nayak et al. [20], a multiagent simulation was developed in ROS (Robot Operating System) in a Linux environment to run experiments comparing the performance of different task allocation algorithms across different communication levels and scenarios. I have built upon this simulation by writing a program that implements this method to improve the task allocation process in the CBAA algorithm. The combination of the CBAA algorithm and the new method is referred to as CBAA_comm. I have also run experiments to compare the performance of the CBAA algorithm with the improved method and the task allocation algorithms described in Chapter 5. The code for the multiagent simulation was written using Python and C++. The simulation consisted of an agent module, a central simulation module and a communication module. The agent module was written in Python and implements the task allocation algorithms that were run in the comparison experiments. It also includes the launch files for running the experiment and the configuration files that contain information on the type and location of targets and the location of agents. The central simulation module was written in Python and implements information about an agent such as its target info list, current bid list in the form of messages sent by the central simulation to the agent. It is important to note that this is still a decentralized system as the task allocation process is performed by all the agents and not by any single agent. The central simulation module also implements the dynamics of the agents and targets and is responsible for generating the plots of the workspace. The communication module was written in both Python and C++ and implements communication between the agents. It implements the Rayleigh-Fading model that was discussed in Chapter 3 and the messages that agents exchange with each other.

The simulation records information such as the distances that agents traveled, the number of messages exchanged, and the tasks completed by the agents in log files which are later analyzed.

6.2. Design of Experiments

In the experiments, various scenarios are generated across multiple communication levels to compare the various task allocation algorithms. A scenario is defined as one instance of choosing a random number of agents, a random number of targets, a random number of adversarial targets, a random set of agent locations, a random set of target locations, and one Rayleigh-Fading power sensitivity threshold.

Targets are modeled as clusters at different locations. This ensures targets are not concentrated in a single region of the workspace and are spread out like in real-world scenarios. The target cluster is modeled as a circular region with a specified radius centered around a location. The number of clusters and cluster radius are generated randomly and used to generate the cluster centers. Targets are then placed inside these clusters based on a uniform distribution. Table 1 provides the constraints on the various parameters of the experiment.

Since this thesis focuses on developing an algorithm for low communication environments, the existing task allocation algorithms are compared to CBAA_comm across six communication levels corresponding to six different Rayleigh-Fading power sensitivity thresholds. Comparison is also done at higher communication levels to determine how well CBAA_comm performs with respect to the existing ones. Table 2 provides the six different communication levels classified into low, medium and high communication and an index representing each level.

Table 1: Ranges of the parameters of the experiment

Parameter	**Range**
Number of agents	Random integer between 5 and 10
Number of targets	Random float in the range [1, 4] representing the ratio of targets to agents. Calculate the number of targets using ratio and round to the nearest integer.
Number of adversarial targets	Random float in the range [1, 3] representing the ratio of adversarial targets to non-adversarial targets. Calculate the number of adversarial targets using ratio and round to the nearest integer.
Number of clusters	Random integer between 1 and 4
Cluster radius	Random integer between 5 and 25
Agent locations	Random integer between 0 and 100 for both x and y locations

Table 2: Communication levels

Power sensitivity threshold	**Communication Index**	**Communication**
-75 dB	6	High
-65 dB	5	High
-55 dB	4	Medium
-45 dB	3	Medium
-35 dB	2	Low
-25 dB	1	Low

The workspace is 100 units by 100 units and is divided into 25 square cells of size 20 units. The origin of the workspace is defined at the bottom left corner. The ship is initialized at the bottom center of the workspace and moves upwards at a constant speed to the top center, after which the simulation ends. Each agent knows the dimensions of the workspace and its location but may not know the location of other agents. The agents do not know the location of the targets and acquire this information by searching for targets and communication with other agents. Figure 6.1 shows the workspace for the ship protection scenario.

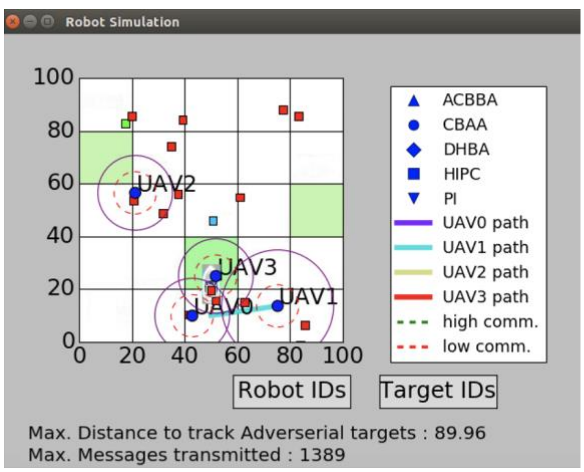

Figure 6.1: Workspace for ship protection scenario

For a single task allocation algorithm, 50 scenarios are generated randomly for 6 communication levels. Six algorithms were compared in these experiments, resulting in 1800 experiments being conducted.

Chapter 7: Experimental Results

This Chapter provides the results of the comparison experiments and identifies different trends and insights in the results for the total number of messages and the two metrics identified in Chapter 3.

As described in Chapter 6, experiments were run to compare the performance of the 5 algorithms described in Chapter 5 with CBAA_comm. There are 3 metrics that are being assessed in these experiments. They are (1) total messages, (2) maximum distance to track adversarial targets, and (3) number of hits to the ship. The results of this experiment are described using box plots and plots of mean values of the following metrics across 6 communication levels for each of the 5 algorithms and the CBAA_comm.

The boxes of the boxplots in Figures 7.1 represents the results of running 50 scenarios for each algorithm and communication level. From Figure 7.1, it can be seen that CBAA_comm has a lower median number of messages sent than CBAA at all communication levels. When compared to other algorithms, it is outperformed by CBBA and HIPC at the lowest communication levels of -25 and -35. However at higher communication levels, it has the smallest median number of messages. The same trend is also observed for the median number of messages. CBAA_comm significantly outperforms CBAA because it restricts communication between agents.

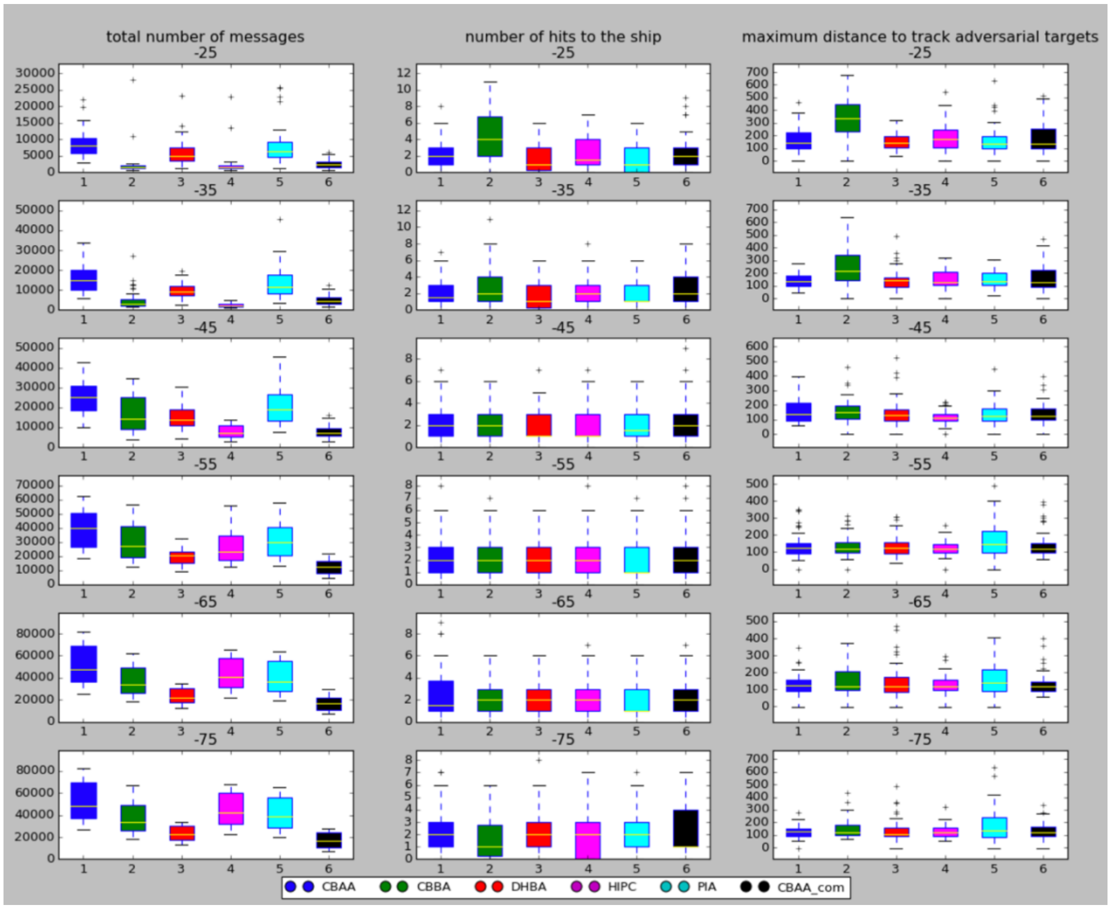

Figure 7.1: Box plots of the total number of messages, number of hits to ship and maximum distance to track adversarial targets across all communication levels when using each algorithm and CBAA_comm.

When the performance of CBAA_comm is compared across communication levels, the median and standard deviation of the number of messages increases as communication improves. This trend can be explained by the higher proportion of messages sent at better communication.

Inspecting the plot of number of hits to the ship in Figure 7.1, CBAA_comm has a comparable median number of hits to the ship to CBAA at most communication levels. However at the highest communication level of -75, CBAA_comm significantly outperforms CBAA in terms of median number of hits to the ship but has a higher standard deviation. Observing the standard deviation in the number of hits to the ship, CBAA_comm has comparable performance to CBAA except at 2 communication levels (-35 and -75). Comparing the performance of CBAA_comm across communication levels indicates that there is no significant improvement in performance.

Finally, inspecting the plots of maximum distance to track adversarial targets in Figure 7.1, CBAA_comm has a comparable median maximum distance to track adversarial targets to CBAA at all communication levels. A similar trend occurs for the standard deviation of the maximum distance to track adversarial targets except at the lowest communication levels of -25 and -35 where CBAA_comm has a higher standard deviation than CBAA. Comparing across communication levels shows that the maximum distance to track adversarial targets decreases as communication improves. When CBAA_comm is compared to other task allocation algorithms, results show that it has a comparable performance.

The similar trends between the number of hits to the ship and the maximum distance to track adversarial targets suggest that the improvement in situational awareness due to estimation compensates for the reduced situational awareness due to reduced communication. Thus the quality of task assignments is not significantly affected. At higher communication levels, restricting communication has a minimal effect on the situational awareness and the improved situational awareness due to estimation dominates and results in better performance than CBAA.

Chapter 8: Discussion

In this Chapter, I look at the impact of communication on the performance of CBAA_comm and compare its performance to that of the CBAA algorithm. Section 8.1 discusses how the performance of CBAA_comm is affected by the quality of communication while Section 8.2 compares the performance of CBAA_comm to CBAA across different communication levels and with different metrics. In both sections, hypotheses are posed for observed trends.

8.1. Impact of Communication on Performance of CBAA_comm

It is observed that the total number of messages exchanged by agents increases as the communication quality improves. The same trend is also observed for the standard deviation of the total number of messages. There are two factors that affect the number of messages exchanged: the power sensitivity threshold and the message threshold value. Given that the message threshold is independent of the communication quality, it is expected that the proportion of messages with a VOC above the message threshold hardly changes. However, as communication improves, the power sensitivity threshold decreases which results in agents receiving more messages. Thus, the number of messages increases.

Results indicate that the number of hits to the ship and the maximum distance to track adversarial targets decrease as communication quality improves. This points to an improvement in the quality of task assignments. Situational awareness characterizes the information an agent has about its environment and other agents and it has an effect on the quality of task assignments. Restricting communication tends to lead to worse situational awareness as agents exchange information at a slower pace with the restriction. On the other hand, estimation improves situational awareness as agents are able to make more informed decisions based on good estimates. As communication improves, the number of messages increases, and situational awareness improves as agents obtain information more quickly and are able to improve their estimates with more information. This subsequent improvement in situational awareness manifests as better task assignments which lead to better performance.

8.2. Comparison with CBAA

The results showing that CBAA_comm sent a lower number of messages at all communication levels than CBAA were expected because CBAA_comm restricts the exchange of messages by defining a threshold value for the VOC of a message.

In the scenarios tested, CBAA_comm has a comparable performance to CBAA in terms of number of hits to ship and maximum distance to track adversarial targets across all communication levels. Comparable performance means that CBAA_comm had a slightly better or worse performance in the number of hits to ship and maximum distance to track adversarial targets than CBAA across all communication levels. This can be explained by considering that communication and estimation are ways to improve situational awareness of agents and thus improve task assignment.

Estimation provides more uncertainty than communication about the state of an agent's environment, including other agents. Given that CBAA_comm employs estimation and reduced communication, the results suggest that estimation is able to substitute for communication in providing information that is used to determine task assignments. It is also important to note that the estimate in CBAA_comm is only used if it has a certain range. This could lead to cases where CBAA_comm performs like CBAA but with reduced communication. It is theorized that these cases happen less frequently as this would tend to lead to worse task assignments due to the reduced communication.

Chapter 9: Conclusions and Future Directions

This chapter contains two sections. Section 9.1 summarizes the results of the comparison experiments and provides conclusions while Section 9.2 discusses ways to extend the current work and other areas that can be explored.

9.1. Conclusion

This thesis presented a new method that is designed to improve the performance of a decentralized task allocation algorithm in low communication environments and reduce the number of messages exchanged between agents in a ship protection scenario. This method was been implemented for the CBAA algorithm and is referred to as CBAA_comm. Experiments were run for a ship protection scenario where the goal is for agents to defend the ship from adversarial targets trying to hit it. The Rayleigh Fading communication model was used to vary communication quality and the performance of this method applied to CBAA was compared against five allocation algorithms : CBAA, CBBA, HIPC, PIA and DH. The performance was assessed using three metrics: number of messages, number of hits to the ship and the maximum distance to track adversarial targets.

Experimental results show that CBAA_comm had a comparable performance in the number of hits to the ship and the maximum distance to track adversarial targets to the CBAA algorithm across most communication levels. This means it performed slightly better or worse than CBAA in these metrics across different communication levels. Comparing CBAA_comm to other algorithms shows that it is outperformed on both metrics by algorithms like DHBA and PIA across most communication levels. When comparing the number of messages exchanged by agents, CBAA_comm sends the least number of messages compared to CBAA and other algorithms. From these results, one can conclude that in the ship protection scenario, estimation can be a good substitute for communication in a ship protection scenario when communication is low or there is a need to restrict communication.

9.2. Future Directions

Future work includes extending this method to a heterogeneous multiagent system with agents and targets of diverse capabilities, implementing a more accurate prediction model to estimate other agent's information and extending this method for other task allocation algorithms. Work can also be done on comparing the performance of this method as the sizes and speeds of agents and targets relative to the size of the workspace increase or decrease.

Bibliography

[1] Waharte, Sonia, and Niki Trigoni. "Supporting search and rescue operations with UAVs." In *2010 International Conference on Emerging Security Technologies*, pp. 142-147. IEEE, 2010.

[2] Nigam, Nikhil, Stefan Bieniawski, Ilan Kroo, and John Vian. "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results." *IEEE Transactions on Control Systems Technology* 20, no. 5 (2011): 1236-1251.

[3] Shaffer, Joshua A., Estefany Carrillo, and Huan Xu. "Hierarchal application of receding horizon synthesis and dynamic allocation for uavs fighting fires." *IEEE Access* 6 (2018): 78868-78880.

[4] Cox, Michael T., and Anita Raja. "Metareasoning: An Introduction." (2011): 3-14.

[5] Russell, Stuart, and Eric Wefald. "Principles of metareasoning." *Artificial intelligence* 49, no. 1-3 (1991): 361-395.

[6] Anderson, Michael L., and Tim Oates. "A review of recent research in metareasoning and metalearning." *AI Magazine*28, no. 1 (2007): 12-12.

[7] Horvitz, Eric J., and John S. Breese. *Ideal partition of resources for metareasoning*. Knowledge Systems Laboratory, Medical Computer Science, Stanford University, 1990.

[8] Pěchouček, Michal, Jan Tožička, and Vladimír Mařík. "Meta-reasoning methods for agent's intention modelling." In *International Workshop on Autonomous Intelligent Systems: Agents and Data Mining*, pp. 134-148. Springer, Berlin, Heidelberg, 2005.

[9] Tang, Su-Yan, Yi-Fan Zhu, Qun Li, and Yong-Lin Lei. "Survey of task allocation in multi Agent systems." *Systems Engineering and Electronics* 32, no. 10 (2010): 2155-2161.

[10] Xuan, Ping, Victor Lesser, and Shlomo Zilberstein. "Communication decisions in multi-agent cooperation: Model and experiments." In *Proceedings of the fifth international conference on Autonomous agents*, pp. 616-623. 2001.

[11] Langlois, Samuel T., Oghenetekevwe Akoroda, Estefany Carrillo, Jeffrey W. Herrmann, Shapour Azarm, Huan Xu, and Michael Otte. "Metareasoning Structures, Problems, and Modes for Multiagent Systems: A Survey." *IEEE Access* 8 (2020): 183080-183089.

[12] Mosteo, Alejandro R., and Luis Montano. "A survey of multi-robot task allocation." *Instituto de Investigacin en Ingenierła de Aragn (I3A), Tech. Rep* (2010).

[13] Brunet, Luc, Han-Lim Choi, and Jonathan How. "Consensus-based auction approaches for decentralized task assignment." In *AIAA guidance, navigation and control conference and exhibit*, p. 6839. 2008.

[14] Zhao, Wanqing, Qinggang Meng, and Paul WH Chung. "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario." *IEEE transactions on cybernetics* 46, no. 4 (2015): 902-915.

[15] How, Jonathan P. *Distributed Hybrid Information and Plan Consensus HIPC for Semi-autonomous UAV Teams*. MASSACHUSETTS INST OF TECH CAMBRIDGE, 2015.

[16] Ismail, Sarah, and Liang Sun. "Decentralized hungarian-based approach for fast and scalable task allocation." In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 23-28. IEEE, 2017.

[17] Wang, Lu, Zhiliang Wang, Siquan Hu, and Lei Liu. "Ant colony optimization for task allocation in multi-agent systems." *China Communications* 10, no. 3 (2013): 125-132.

[18] Patel, Ruchir, Eliot Rudnick-Cohen, Shapour Azarm, Michael Otte, Huan Xu, and Jeffrey W. Herrmann. "Decentralized Task Allocation in Multi-Agent Systems Using a Decentralized Genetic Algorithm." In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3770-3776. IEEE, 2020.

[19] Herrmann, Jeffrey. "Data-driven metareasoning for collaborative autonomous systems." (2020).

[20] Nayak, Sharan, Suyash Yeotikar, Estefany Carrillo, Eliot Rudnick-Cohen, Mohamed Khalid M. Jaffar, Ruchir Patel, Shapour Azarm, Jeffrey W. Herrmann, Huan Xu, and Michael Otte. "Experimental comparison of decentralized task allocation algorithms under imperfect communication." *IEEE Robotics and Automation Letters* 5, no. 2 (2020): 572-579.

[21] Otte, Michael, Michael J. Kuhlman, and Donald Sofge. "Auctions for multi-robot task allocation in communication limited environments." *Autonomous Robots* 44, no. 3 (2020): 547-584.

[22] Sujit, P. B., A. Sinha, and D. Ghose. "Multi-UAV task allocation using team theory." In *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1497-1502. IEEE, 2005.

[23] Dai, Wei, Huimin Lu, Junhao Xiao, and Zhiqiang Zheng. "Task allocation without communication based on incomplete information game theory for

multi-robot systems." *Journal of Intelligent & Robotic Systems* 94, no. 3 (2019): 841-856.

[24] Cheng, Peng, and Vijay Kumar. "An almost communication-less approach to task allocation for multiple unmanned aerial vehicles." In *2008 IEEE International Conference on Robotics and Automation*, pp. 1384-1389. IEEE, 2008.

[25] Estefany Carrillo, Suyash Yeotikar, Sharan Nayak, M. K. M. Jaffar, Shapour Azarm, Michael Otte and Huan Xu, `"Communication-aware multi-agent metareasoning for decentralized task allocation." Univ. Maryland, College Park, MD, USA, Tech. Rep., 2020.

[26] Zheng, Yahong Rosa, and Chengshan Xiao. "Simulation models with correct statistical properties for Rayleigh fading channels." *IEEE Transactions on communications* 51, no. 6 (2003): 920-928.

[27] Johnson, Luke, Sameera Ponda, Han-Lim Choi, and Jonathan How. "Asynchronous decentralized task allocation for dynamic environments." In *Infotech@ Aerospace 2011*, p. 1441. 2011.