7-2017

Investigating Diversity in Open Multiagent Team Formation

Pooja Ahuja
University of Nebraska-Lincoln, ahuja.pooja22@gmail.com

INVESTIGATING DIVERSITY IN OPEN MULTIAGENT TEAM FORMATION

by

Pooja Ahuja

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Leen-Kiat Soh

Lincoln, Nebraska

July 2017

INVESTIGATING DIVERSITY IN OPEN MULTIAGENT TEAM FORMATION

Pooja Ahuja, M.S.

University of Nebraska, 2017

Advisor: Leen-Kiat Soh

Team formation is the most rudimentary form of interactions in distributed AI and multiagent systems as it allows coherent collections of agents to work together in a beneficial manner towards a common goal of interest. Basically, individual expertise are assembled together in an additive fashion for accomplishing tasks together. A plethora of the related studies found in the literature often make several unrealistic assumptions such as coordination amongst the agents, or agents having knowledge of the whole environment, or agents and/or tasks are of the same kind, or a static environment setting. Against this background, we argue that there are real-world characteristics that make team formation more challenging: (1) There is no or minimal pre-coordination since storage and retrieval is a costly affair, (2) There is diversity amongst types of agents (Apprentices, Generalists, and Specialists) and tasks (Low, Medium, and High), (3) The environment is open i.e., agents and tasks can leave and enter the environment, and (4) Agents are continuously learning and improving their capabilities.

The main contribution of this research is to study in great depths the impacts of various permutations of open and diverse environments on team formation and how agents learn to form these teams. Based on the findings of these studies, we demonstrate that both diversity and openness have impacts on the team formation. Having evaluated the results of the impacts of openness and diversity on the environment we, to strengthen the robustness of the original model, we introduce an enhanced version of this model.

The next contribution of this thesis is putting forth an enhanced probabilistic modelling solution. To be able to carry out new investigations and introduce the new model, we have restructured and cleaned up the simulation software used for building the original model. Having implemented the enhanced model, we then show how this new model performs better than the original model. The final contribution of this thesis was to show why the new model performed better than the original model.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Professor Leen-Kiat Soh. I appreciate all his contributions of time, patience, and ideas to make my graduate study experience productive and stimulating. Words fall short to thank him enough for all the energy and time he has invested to help make me a better researcher. He is one of the kindest and most patient advisor there can ever be. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the Masters pursuit.

I would like to thank my thesis committee members Professor Stephen Scott and Professor Ashok Samal, for all of their guidance throughout this process; their discussions, ideas, and feedback have been absolutely invaluable.

My appreciation also extends to my lab-mates, Elliot, Bin, and Dr. Eck for their helpful discussions and willingness to help me throughout my research.

In closing, I would like to thank my family and friends without whom this research would not have been possible. I thank them for all their love, patience, encouragement, and memes which helped me stay focused and positive.

Dedication

To my parents.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Team Formation

Human beings show natural traits of working in teams towards accomplishing a common goal. Real-world environments foster upon skilled organizations that work towards performing different duties as a team. The performance of such an environment depends on both the individual level as well as the team level efficiency and effectiveness. Effectiveness is the measure of the agent's ability and skillset to perform, the task assigned and efficiency is the measure of how well the task was solved and the rewards that were earned. Individual and team-level efficiency and effectiveness are the measure of rewards earned and the measure of the abilities respectively at agent and team level, respectively. This realm of the real world can be extended to artificial societies through incorporating it in multiagent systems. Thus, team formation amongst agents is one of the most researched areas (Ray and Vohra, 2014); Procaccia and Rosenschein, 2006). Given this background, there are many applications of team formation from the real world that are mapped to a multiagent environment, for example, search and rescue operation, teams playing soccer, the predator prey domain, etc. There is a plethora of literature in a wide spectrum that studied the domain of team formation in multiagent systems, ranging from agents in teams learning by observing human users (Oblinger et al. 2006; Rybski et al. 2008), applications that learnt algorithms from observational models (Debhi, et al. 2012), teaching robots to solve puzzles, (Lee et al. 2013), learning to fly (Isaac and Sammut

2003; Sammut et al. 1992), driving a simulated tank in formation (Fernlund et al. 2009), clustered data (Chaimontree et al. 2012).

1.2 Challenges

Horling and Lesser (2005) defined a set of key characteristics of a team as follows:

> "Teams in general are goal-directed and short-lived; they are formed with a purpose in mind and dissolve when that purpose no longer exists, or when they cease to suit their designed purpose, or when the profitability is lost as agents depart."

The three primary functions comprising the team formation process include, (1) an agent finding a suitable task, (2) team formation and (3) execution of the task. There are environmental factors that make it difficult for an agent to decide which tasks are suitable, and how an agent should decide which tasks are better than the others if there are multiple suitable tasks. There are also issues with team formation when agents have to be assigned to a task for a task to be successfully completed. In particular, we see four key challenges: (1) forming teams with no or minimal pre-coordination, (2) forming teams under open environments, (3) forming teams under diverse environments, and (4) forming human teams where human learning is present. These challenges are especially significant when human teams are involved.

No or Minimal Pre-Coordination. Most of the previous works in the field of team formation in multiagent systems assume that agents have knowledge about the other agents, or they know the environment really well beforehand. There is a good amount of research in literature which studies team formation with coordination under multiagent systems (Shehory et al. 1998b; Shehory and Kraus 1998a; Brooks and Durfee 2003; Caillou et al. 2002). However, such assumptions are somewhat impractical in cases when, information is costly to be conveyed across agents, or the agent population is high and it is not possible for every agent to store information about the others, or maybe the environment is dynamic and new agents keep joining and leaving the environment, especially so in human teams. No or minimal pre-coordination makes team formation challenging because agents have incomplete information of the environment. For example, agents are not aware of the other agents in the environment or their expertise and capabilities, hence they are not sure which tasks to bid on or who their perspective teammates could be. Thus, no or minimal pre-coordination makes teams formation more complex.

Openness. Other works (Shehory and Kraus 1998; Liemhetcharat and Veloso 2001) which have not based their work on the assumption that every agent is well-versed with the formation information, assumes that the environment is not open (static) and thus no new agents or tasks leave or enter the environment. However, this is not always the case in the real world. We see new humans and tasks leave and enter an environment. For example, in a company, employees leave and new ones join, or the company might open a new branch which deals with tasks that the company had never seen before. Openness causes the expertise in the environment subject to fluctuation, thus in return affecting the decision making of agents. For example, in a software development scenario, an effective software coder leaves the team, the entire team might get a negative hit along with the project. Or if a certain task that a team was good at happened to get absorbed then the team will have to start over and learn a new task, rendering the present skillsets useless. On the other hand, if a new task enters the environment, then the agents will be motivated to learn and improve their capabilities to be suitable for the new task. Thus, openness causes the agents to be uncertain of the environment as they are not sure of the agents and tasks being available in the environment. This uncertainty also makes it difficult to optimize the long-term expected utility while forming teams. This is how openness is a challenge to form teams on account of the uncertainty it introduces.

Diversity. Most prior works deal with a non-diverse (homogenous) set of agents and tasks Albrecht and Ramamoorthy (2012), they assume that all agents have the same capabilities or show the same traits and the tasks are of similar kind as well. However, we do not see this in reality. All human beings have different levels of capabilities, for example, a cook is good at cooking but need not be good at coding. The tasks that humans work on are different, some of them are very easy like writing documentation for a codebase, and some of them are relatively harder, like creating a system of agents to help human beings form teams. And not all the tasks require the same type of expertise either, tougher tasks may require more experienced and skilled personnel as compared to an easier task. Diversity amongst agents helps the agents learn a variety of capabilities through a diverse set of agents and a diverse set of tasks presents every agent a more varied opportunity to be able to get a task assigned. Diversity along with openness in the environment makes team formation even more complex. For example, when there is high task diversity and low agent diversity along with high agent openness, it might be difficult for existing agents in the environment to cope with the wide variety of tasks available. Also on account of high agent openness, when some agents do start getting the hang of tasks present in the environment, there is a high chance that they might exit from the environment. Whereas on the other hand in the same scenario if there was high diversity amongst the agents as well then, the agents would have a wider variety of capability to deal with the high diversity of tasks present in the environment, as well as teach their peer agents how to get better at tasks. Thus, diversity along with openness makes the environment complex and team formation more complicated.

Human Learning. While there exist numerous works that model their agents to learn the behaviors of the other agents based on past observations and interactions (Abdallah and Lesser 2004; Sun 2001; Chalkiadakis 2007; Jiang et al. 2008) and incorporate learning into team formation, few take into consideration environments where there will be human presence. When humans are present in the environment, they inevitably tend to learn from each other or they keep enhancing their skills as and how they keep performing tasks, also humans teach each other as well. Human learning is a challenge because it is hard to model. Also with learning humans tend to improve. With this improvement in the agents through learning, there is uncertainty in the environment due to incomplete information of these changes. This makes it more difficult to model the potential teammates that could work on tasks together. Also by introducing human learning it might become complex to form optimal teams in the long run since the agents will keep getting better on account of the learning.

1.3 Proposed Approach

To overcome the assumptions and drawbacks in the previous research works that have encroached the domain of team formation in multiagent systems, Chen (2017) proposes an autonomous and open environment in a multiagent environment for team formation and we extend this model to introduce diversity. To summarize the model Chen et al. (2015), the term autonomous refers to the ability that agents do not have any information about the other agents in the environment, there is autonomy amongst agents, they are only aware of the tasks in the environment. Openness is the measure of new agents and tasks joining and leaving the environment. Diversity is defined as the spectrum of types of agents and tasks available in the environment. Along with a realistic environment the model also equips the agents with learning and modelling uncertainty in task accomplishment. There are two types of learning Chen et al. (2015) – learning by observation and doing. Learning by observation helps us reduce the time it would take an expert agent to teach a novice agent, since the novice agent can simply learn by observing the expert agent and improve. Learning by doing helps agents improve their capabilities by putting their skillsets to execution. Modeling of the task assignment is needed because of the openness in the environment. There is no guarantee that a task or an agent present in the environment will be present in the future as well or new tasks and agents may enter, changing the dynamics of the environment. The probabilistic modeling helps the agent realize the chances of it being assigned to a task and also there being sufficient teammates for the task to be executed successfully.

Since this model Chen (2017) mostly focused on the openness in the environment, we extended this model to include diversity as well. As seen from the challenges in section 1.2, adding diversity to an open environment complicates the team formation further. After having simulated the model to handle diversity as well, we analyze the environment at finer details. Having carried investigations at minute levels for both diversity and openness we realized that we could fine tune the probabilistic modeling Chen (2017) to further enhance its robustness in an open and diverse environment. We realized that with Chen (2017), probabilistic modeling the agents are in a continuous chase of finding a task with the right number of teammates. This leads to fewer tasks being auctioned off, since even though there is the right amount of expertise available in the environment, they are all split over different tasks rather than being channeled on common tasks. To overcome this, we improve the probabilistic modeling to let the agents also be aware of the possible number of teammates a given task can expect. This now helps the agents to bid for tasks which are more likely to have the right number of teammates rather than chasing the teammates around.

With the new enhancements, we now carry out further investigations to analyze the impacts of diversity and openness on this new enhanced approach. A series of investigations helps us realize that the proposed enhancement does better than the original model.

1.4 Contributions of this research

The primary goal of this thesis focuses on analyzing the impacts of agent and task openness and diversity on the robustness and dynamics of team formation and the environment. For this, several analyses were conducted with different permutations and combinations of diversity and openness to analyze at fine levels what the impacts of these could be on the environment. Along the way, we also enhanced the simulation software system to allow for configuration and simulation of task and agent diversity in the environment.

After having deeply studied the impacts of diversity and openness on the environment and the team formation, we identified several key relations between diversity and openness. Upon further investigations, we realized that the model's robustness could be enhanced by fine tuning the probabilistic modeling further.

The next contribution of this research is to enhance the model Chen (2017) to tackle diversity along with openness and help agent to better probabilistically model their environment and hence contribute better to the environment.

Having implemented an enhanced version of the approach, we conducted a series of comparison tests to prove that the new improved approach works well and better than the original approach Chen (2017). We also conducted experiments to find the impacts of diversity and openness on the environment based on this new approach.

1.5 Overview

- **Chapter 2** situates this thesis in the literature in this related field and sheds light on the drawback of some of these works and the need for our research.

- **Chapter 3** gives a summarized account of the autonomous and open team formation model approach proposed in Chen (2017). It also describes our extension of this model to introduce diversity in the environment.
- **Chapter 4** presents the analysis conducted to evaluate the aforementioned approach to better understand the impacts of diversity and openness on the environment and the results obtained from these findings.
- **Chapter 5** puts forth the new proposed solution which will help enhance the probabilistic modeling approach for agents to better form teams.
- **Chapter 6** describes the implementation details.
- **Chapter 7** finally concludes this Thesis and emphasizes on the implications made from our investigations and the key findings of this research.

The recommended reading order is as shown by the arrows in Figure 1.1.

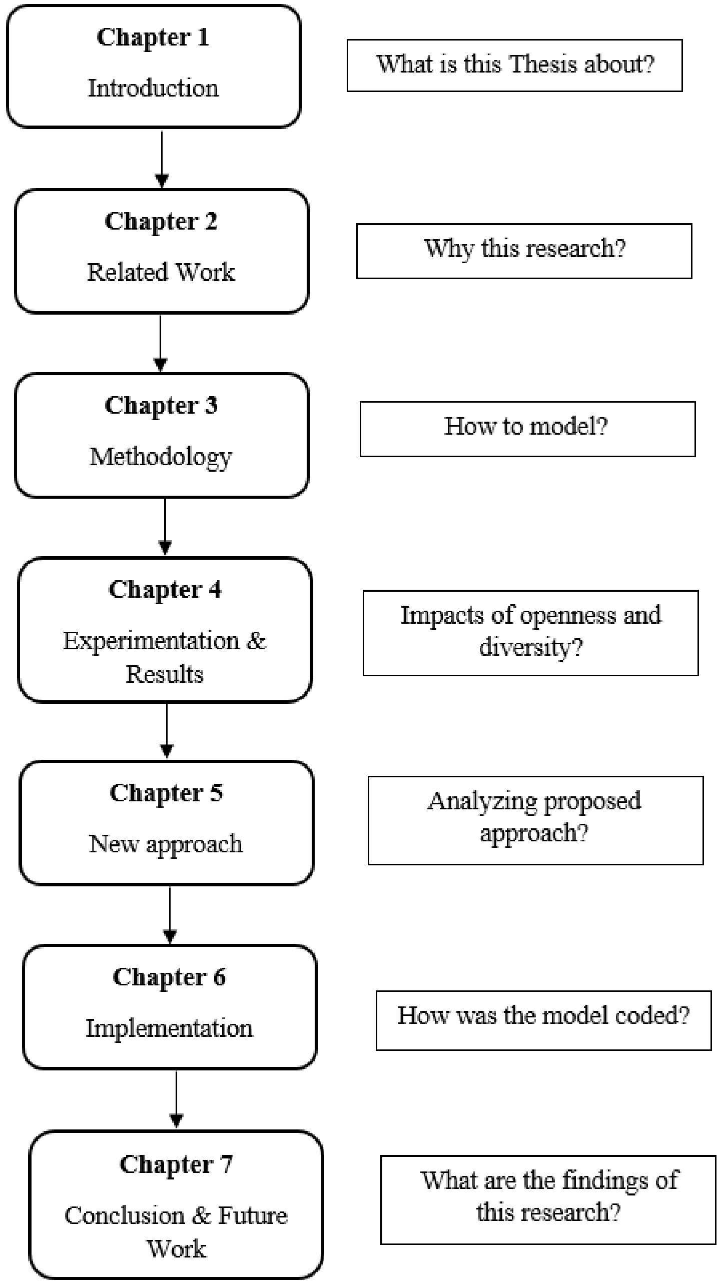


Figure 1.1. Thesis Structure Flowchart.

Chapter 2

RELATED WORK

An extensive literature review helps us understand what work has been done in the human team formation. We are particularly interested in team formation in a **diverse** and **open** environment. We equip our agents with **human learning** by *doing and observation.* The model Chen et al. (2015) considers a **tradeoff between learning and task accomplishment**. Our Related Work section has the following subsections: 2.1 Openness, 2.2 Diversity, 2.3 Human Learning and 2.4 Tradeoff Between Learning and Task Accomplishment. Through this literature review we have realized that very few research works incorporated both agent and task openness, and diversity. There is very little literature about the impacts of varying agent and/or task population and/or skillset on the coalition[1] formation and the rewards earned. We have also realized that few works took into consideration human learning. Also, there are few papers that dealt with the tradeoff between learning and task accomplishment in open environments.

This related work focuses on how coalition formation could be carried out in environments with task and agent openness and diversity. *Trading-off between learning and task accomplishment in open environments is one of the key contributions of our work alongside varying both – agent and task, diversity, and openness.*

[1] We use coalition and team interchangeably throughout this chapter.

2.1 Openness

2.1.1 Agent Openness Only

There are some research approaches that addressed coalition formation in an agent-open environment.

For example, Jamroga et al. (2013) and Shehory et al. (2000) described agent openness similarly to ours where agents could leave and enter environment. Yokoo et al. (2005) proposed a theoretical algorithm for mapping coalition formation in an open and anonymous environment. Agent openness was a measure of the leaving and entering of agents in the environment. Agent openness was introduced theoretically in the aforementioned works and no experimental results were provided to investigate its role or impacts. In another work, Maret et al. (2004) implemented openness in societies of agents. Klusch and Gerber (2002), in their research, modeled coalition formation in a dynamic environment, where agents can enter and leave the environment at any point. There was another study that proposed a trust-based coalition formation amongst agents by Griffiths and Luck (2003) where the authors considered their environment to be dynamic, uncertain, and noisy. The uncertainty in their environment came from the ability of the agents to join and leave the environment. Another work, by Pinoyl and Sabater-Mi (2011), developed an open environment that helped their agents search and work with reputable agents. In addition, Huynh et al. (2006) allowed their agents to assess the quality of its peers' performance in an open environment.

However, despite having modeled coalition formation in an agent-based open environment, these aforementioned approaches did not address task openness or the impacts of varying the amounts of agent openness on agent performances. In our work, we implement both agent and task openness, as well as studied the impacts of varying the degree of the openness on the environment as well as the rewards earned. Our approach helps us hypothesize how both agent and task openness have an impact on the rewards earned and the coalition formation. It also helps us understand how different degrees of openness affected the coalition formation. In a way, the aforementioned previous work can be integrated with our model of task openness to enrich their environment and can also be extended by varying the amount of both agent and task openness to study how their model performs subject to varying degree of openness.

2.1.2 Both Agent and Task Openness

Next, we point out two approaches that are significant to us because of their similarity to our work. Both these approaches implemented both agent and task openness.

First, Shehory and Kraus (1998) defined their agent openness as the ability of agents to appear or disappear from the environment, and their task openness was defined as the introduction of new tasks and removal of tasks that were already present in the environment. Their environment is a dynamic and open framework, similar to our work. They used RETSINA, an existing web-interface to implement the agent and task openness in the dynamic environment. However, despite this work and our work modeling a similar environment, dynamic and open, there are a few striking differences between both works. In this work, if a new agent came into the environment it announced itself to the other agents present in the environment and exchanged information about itself with them, this helped in the coalition formation. In our case, agents do not communicate with each other and the central auctioneer was responsible for the coalition formation. Their work may not be extended to scenarios where autonomous agents (no communication between agents) are driving the environment, unlike our work. Each time a new task arrived in the environment, the coalitions were re-calculated for these new tasks, the most beneficial coalition was then chosen, in this work. In our setting, coalitions are not re-formed after the auction has taken place. We model different degrees of both agent and task openness and studied their impacts on tasks and the rewards earned. This helps us formulate and test the hypothesis that dealt with an open environment and the degree of this openness. However, their work did not account for varying degree of openness.

Second, we acknowledge the work by Jumadinova et al. (2014). They used two openness metrics (agent and task openness) to model the dynamic nature of both the agents and the tasks, in a search and rescue operation scenario. They defined agent/task openness as the appearance of new agents/tasks in the environment. They justified the need for both agent and task openness in an environment by arguing that, not every mission will have the same set of tasks, there may be new tasks entering the environment, when working in a team for a search and rescue mission, the same set of agents might not appear again for some other mission, there may be new agents that join the environment. They implemented both agent and task openness as a fraction metric of new agents or tasks entering the environment at the end of each time step. They also studied the impact of openness on multiagent learning and teaching model. Note that we extended the notion of user and task openness from (Jumadinova et al. 2014; Chen et al. 2015). However, unlike Jumadinova et al. (2014), we do not restrict openness to only define the rate at which agents/tasks enter the environment. We extend openness to be a measure of new agents/tasks *entering* as well as *leaving* the environment. This is a key difference, in the way both approaches model their openness. We could integrate our openness into their model, as that would help make their model more realistic by not limiting their openness to agents/tasks only entering the environment with no provisions of leaving. While Jumadinova et al. (2014) investigated the impact of openness on agent performance, the authors did not consider the potential complications caused by diversity in tasks and agents, unlike our work. Also, it treated its human learning as a black box and did not model its learning process formulaically, unlike our work.

2.2 Diversity

2.2.1. Agent Diversity Only

There are approaches that modeled agent diversity in the environment for agent coalition formation.

For example, Shehory and Kraus (1998) defined agents like our setting—each agent has a capability vector. They defined diversity as different classes of agents, where each class performed specific tasks (they introduced, interface, task, and information agents in their environment for simulations). Amongst the same class of agents too, each agent had a different capability vector which differentiated it from the other agents during considerations for the execution of tasks.

In another study, Liemhetcharat and Veloso (2011) talked about diversity of agents. They defined diversity by introducing roles (e.g., for simulations they mapped a soccer scenario where the roles were defined as, defense, midfield and offense players). Further agents amongst the same role had different level of capabilities. Each agent had different levels of capabilities.

Chalkiadakis and Boutilier (2008) modeled agent diversity similar to our work. In their model, they considered agents of different professions such as carpenters, electricians, and plumbers. Depending on how different the capabilities of a pool of agents within the same profession was, this work clustered their agents as good, medium, and bad. This consideration for diversity was like ours. In our model, agents had different sets of capabilities, and for each capability, there was a level of expertise associated. For example, agents who had high level of skills for some of their capabilities, are experts. However, the emphasis of diversity in this work and ours differed.

However, there are three key differences between the aforementioned research approaches and ours. These approaches (1) did not consider the effects of varying the diverse agent population on coalition formations or the rewards earned, (2) modeled only agent diversity and did not account for task diversity, and (3) defined agent diversity as the clustering of agents based on their professions/roles. First, we carry out our experiments to test the impact of the diversity of the agents on coalition formation by varying the population of our agents (more generalists or more specialists at a given time). This diversity variance allows us to study the real-world human coalition formation more accurately as diversity plays an important role in the human world (Marcolino et al. 2013). This helps us record hypothesis for testing impacts of different types of populations of a given skillset and their effect on the environment and task accomplishment. This also makes our environment more realistic. Second, unlike our work, the aforementioned works, did not account for task diversity. We have different types of tasks available (Low, Medium, and High) depending on the capabilities required to accomplish these tasks. It would be nice to see them account for task diversity, they assume no diversity amongst these tasks. Task diversity helps us replicate the real-world scenario closely. Third, the way the aforementioned approaches and our work define agent diversity differs. These works divided their agents based on professions/roles, amongst these classes the agents had different capability vectors. For these approaches, agent diversity is presence of different class of agents based on their profession with varied capabilities in the environment. Agent diversity in our investigation is the measure of how different the capabilities of the pool of human users is. We conceptualize agent diversity by introducing three types of users: generalist, specialist, and apprentice. Generalists are described as having a moderate skill level in a moderate number of capabilities. Specialists are described as having a high skill level in a small number of capabilities. Apprentices are described as having low skill levels in all capabilities. Following this definition, a set of users of all three user types is more diverse than a set of users with only one or two user types. Users of each type have differing levels and numbers of capabilities, all drawn from normal distributions specific to their type.

2.2.2. Agent Diversity and Task Diversity

There are some interesting approaches that modeled both agent and task diversity together. For example, Albrecht and Ramamoorthy (2013) presented their work on coalition formation without any pre-coordination between the agents. They formalized their framework as a Bayesian game where the behavior of the player was determined by its type. They tested their framework in a foraging domain. They implemented agent and task diversity, in the form of variations in the abilities of the agents and the level of agent ability required various tasks associated with a food item. The impact of diversity on team formation was studied by Van de Vijsel and Anderson (2004). They defined agent diversity as the presence of heterogeneous agents (different capabilities) in the environment, and task diversity as the presence of a number of different tasks (requiring different agent capabilities) in the environment.

However, both Albrecht and Ramamoorthy (2013); Van de Vijsel and Anderson (2004) did not carry out experiments to show any link between varying agent and task diversity and nor did they consider openness in their environment, unlike our work. In our model, we take into consideration both agent and task diversity. We also carry out experiments to study the impacts of varying agent and task diversity on the environment as well as if there exists any link between agent and task diversity and openness.

2.2.3. Variations of Agent Diversity or Task Diversity

There were a few interesting approaches which focused their study on analyzing the impacts of varying either agent or task diversity on the environment and on coalition formation, similar to the interests of our work.

For example, Campos and Willmott (2004) modeled agents, similar to our research, they also examined the impact of agent diversity on the environment by varying the population of these diverse agents. They considered a diverse set of agents depending on the level of expertise of their capabilities, similar to our model. They further created environments with varying number of expertise available in the environment. They ran experiments with pure populations of 100% of each strategy and 50:50% ratios of each combinations of agents.

There was another research which focused on varying the task diversity and its impacts, Kraus et al. (2003); Kraus et al. (2004), proposed a heuristic coalition formation method in a “Request for Proposal” domain. They accounted for task diversity by introducing two types of tasks, specialized tasks, and regular tasks. These are distinguished based on the probability of being assigned to an agent. Specialized tasks had a smaller probability of assignment than regular tasks. During their experimentation, they defined 50% specialized tasks and the others as regular tasks.

On the other hand, in our research we defined both agent and task diversity variations, unlike Campos and Willmott (2004) who defined only variations in agent diversity and Kraus et al. (2003); Kraus et al. (2004) defined only variations in task diversity. In our work, agent diversity is defined based on the threshold of the capabilities an agent had (Apprentice, Generalist, and Specialist). Task diversity is classified as the level of variety between tasks present in the task pool (Low, Medium, and High diverse). This diversity helps us capture the real-world scenario and helps us evaluate how diversity could possibly impact the coalition formation and the learning process. Even though both models our work and Campos and Willmott (2004) modeled varying agent diversity population, both works however used this for different analysis. We utilize it to answer the impact of population diversity on the environment, coalition formation, rewards earned and the bids made. We also analyze it to form hypotheses about openness and diversity, however Campos and Willmott (2004) did not account for openness. On the other hand, Campos and Willmott (2004) utilized the varied population split and analyzed its effects on the dynamics of the coalition: when did agents leave a coalition, when did agents join a coalition, etc. Unlike our work, this work did not account for task diversity. They only had the same task over all the episodes (simulation runs). It would be interesting to see experiments implementing their model of task diversity as well. Then interesting questions relating to relationships between task and agent diversity could be analyzed. As for the work by Kraus et al. 2003; Kraus et al. (2004), it is possible to extend it by implementing our agent diversity model. This would help investigate how good their model is when subjected to agent diversity and investigate any relationships between agent and task diversity.

As seen from all the works mentioned in 2.2, we realize that there have been very few research efforts that incorporated *both* agent and task diversity, together. There have been even fewer studies that investigated the impacts of varying both the agent and task populations in the environment. However, our work integrates *both* agent and task diversity. We also carry out experiments to study the *impacts of varying this diversity* on coalition formation and the rewards earned.

2.3 Human Learning

2.3.1 Learning by observation

Through our literature review we realize that a few research approaches that modeled their agents with learning by observation.

For example, Moura and Sarma (2005) implemented an imitation process in multiagent systems. The agent was provided the context of the environment and the actions performed by its peers, at regular intervals. These contexts helped the agent imitate the actions of its teammates and then worked towards achieving the goal. They executed their imitation learning through two agents: the smart and the ignorant agent. The smart agent was provided with all the knowledge of the environment beforehand whereas the ignorant agent was provided with *only* some knowledge of the whole environment. The ignorant agent learned by observing the behaviors of the smart agent. They implemented the imitation process through a process they called imitation algorithm. They allowed their agents to observe the state of the environment and the public state of other agents, and the actions executed by other agents. The smart agents willingly let the ignorant agents observe their public information. This work could be extended to include probabilistic modeling to model the probability of the actions of an agent on the environment, rather than just viewing the present state of the environment. The learning gain in this work was accomplished when addition of new useful knowledge to the ignorant agent's knowledge model. This work did not involve uncertainty in the environment caused by openness.

Collins et al. (1991) presented their work on learning through apprenticeship. The teacher showed the apprentice how to execute a task in steps, starting from easy steps. Then as the apprentice learned, the teacher further showed the student how to execute higher levels of the task, this process of learning through observation in phases was carried out until the apprentice became capable enough to execute the task on its own. The learning gain metric is the learning of the apprentice through the observation of teacher, in the form of the teacher doing actions and the apprentice noting the actions and the consequences. It would be interesting to see how would this model perform in presence of openness. What happens in during the teaching process the apprentice left the environment or immediately after learning? This would be a waste of time for the teacher. Or what if the teacher left in between the process of teaching? We could extend our probabilistic model to this work to see how it could enhance their model.

Van de Vijsel and Anderson (2004) modeled learning through observation as well. An agent learned about the new attributes of the surrounding agents (their actions and the corresponding consequences) through observation and started formulating coalitions when it was advantageous to all concerned thus allowing stability for the environment. An agent modeled its environment (by observation) for every coalition, this helped it form perspectives about the agents it worked with and also, the coalition that it had joined. Through observations, an agent learned the information about the peers it worked with by learning the actions taken by the peers and also the consequences of the actions. It also learned about the environment by learning the coalitions an agent worked with and the outcomes of this coalition formation. This way by constantly updating its beliefs, it learned more about the other agents' attributes and the consequences of their actions. By constantly updating an agent's beliefs through observation (of teammates and environment), the learning gain increased in the form of gained beliefs. Like the previous works, this work has not accounted for openness in the environment.

In another study, Schatten (2014) applied learning by observing to smart residential buildings. The author equipped the agents to learn by observing the actions of other agents as well as human-beings in the residential building. Such observation is modeled in the form of waiting for actions to take place and then add this updated actions and consequences to its own knowledge base. They implemented this observational learning in the form of observer behavior. The observer waited for actions to take place, and then added additional rules and facts to the knowledge base. They also provided an example, where a newly installed speaker should be able to adjust its volume level by observing the volume of the other speakers that are already installed in the household. This work did not integrate openness. The learning was a measure of the addition of new rules and facts from observations of new tasks to the knowledge base of an agent.

Barrett et al. (2013) studied coalition formation in the pursuit domain. They implemented generalized learning and transfer learning methods that in return helped their agents to better model the behavior of their present peers to be able to predict the models of the unseen future peers. The agents observed the action and consequence of their peers and modeled this. The agents build a model of all the peers (through observation) that they have worked with and store it offline, when they go online there is a chance they may encounter new teammates during the coalition formation. These models are an action-and-consequence map, of the peers, which was observed by the agent. Agents refer to these offline models when forming a new coalition. These models equipped the agents with the knowledge of their peers beforehand and accordingly the agents select actions which will be beneficial. However, the agent selected from all the offline models, assuming that the past teammates are a representative of the future teammates. They implemented Transfer Learning using TrAdaBoost, that uses boosting to learn models of agents. However, the generalized learning could get complicated on introduction of openness, it will raise the question of how many models to store. It would be interesting to extend their model and include openness and test our probabilistic modeling on their work, to test to see how the learning gets affected in presence of dynamic agents. The learning gain was directly related to the addition of new learnt models through observation.

Floyd and Esfandiari (2011) developed a framework which enabled their agents to learn through observation, in a partially-observable environment. An agent observed an expert agent and its interactions with the environment as an observer. The observer agent observed the expert agent over a period while the expert interacted with the environment. The observer observes the actions performed by the expert and the responses it received from the environment for it. It then learned to perform the same task as the expert agent and updated its training model to perform better. There was no openness considered in the environment. There are interesting questions that could be answered with openness, what happens if an expert left the environment while the observer was learning? Does the observer reset the knowledge it gained? Or does it figure out the rest of the actions on its own? The learning gain for this work is the knowledge gained by the observer from observing the expert agent.

Johnson and Gonzalez (2013) built a model where a group of agents observed or imitated another group of agents while they performed a task. They implemented this is a Collaborative Context Based Reasoning framework. One group of agents learned from another group only if they were working on the same task. They used behavior maps and case based reasoning to employ learning by observation in their framework. These behavior maps are built on the observed expert team. The learning team then uses these behavior maps to plan their next actions. This work did not account for openness. Learning gain for the observing team is the addition of new knowledge achieved through observation of the expert team. Learning gain is the addition of behavior maps.

There were some studies, that presented their findings on learning of probabilistic human-user models. Dillenbourg (1991), designed a learning strategy that helped agents to understand the dynamics of their environment. The agents learnt the models of their assigned human counterparts, and hence better chose and performed tasks with efficiency. In other research works Greer et al. (2001); Vassileva et al. (1999); Vassileva et al. (2002); Vassileva et al. (2003); Bull et al. (2001) implemented models wherein their agents learnt the probabilistic models of their assigned human-users and communicated this information amongst themselves. After having exchanged this information the agents formed coalitions to assist their assigned human-users.

2.3.2 Learning through reflection

There are a few approaches that have presented their work on learning through reflection, analogous to an agent observing its own processes and reasoning about them to improve its performance.

Wu et al. (2011) incorporated learning through self-observation amongst agents. Their algorithm helped agents observe the recent history of interactions as well as the utility gains from these interactions. Agents took into consideration this recent history of interactions (which is limited to a certain amount) and chose their next actions. This selection of the next action was based on the probability of optimality. Where an optimal solution was an action that led the agent towards gaining optimal rewards. Their framework gave the agents the liberty to plan their own coalition. However, the algorithm did not consider the learning capabilities of agents at any point of time in the history. The environment is assumed to be static and no agent leaves or enters the environment. The learning gain is the addition of new histories of interactions.

In another approach, Schon (1987) studied about human-learning through reflection from a phycology's perspective. A human re-thought the solution carried out by himself or herself to execute a task successfully. The human then reflected upon this re-thinking process and learned through it. Thus, in this case, the observation was done by the person who carried out the task of his or her own work. Schon argued that a person truly learns when he/she starts to evaluate his/her own actions with a critical eye. The rethinking began with the recognition of a doubt. This process of "catching oneself" was important for reflectivity. These doubts are best clarified through eliminating any actions that lead to the doubt and then trying to rectify them. Then after the settlement of the doubt, people affirm their knowledge of the situation. Now they can deliberate about if they carried out the right action for the given situation, by answering the questions: what do I do? How did I do it?

How does it affect others and me? For our future work, we could integrate this model of re-thinking in our agents.

2.3.3 Learning by doing

Learning by doing has also been modeled in a few research approaches.

For example, Inaba et al. (2000) studied human learning in collaborative setting. Their studies showed that humans were capable of learning through the experiences gained by doing and the learning gains increased with completion of each task. The agents came up with a common learning goal and formed coalitions. The agent that had the highest capability then performed the task and the other agents learned by performing the same actions, the authors however did not dive deep into the implementation details. The authors suggested that, when performing tasks in a collaborative setting, humans learned through the performance of the tasks and there was also a change in their behavior as and how they learned through the experiences of doing those tasks. Learning gain the knowledge gained by doing the same tasks as done by an expert agent. There was no openness considered in the environment.

Steinhage and Bergener (2000) implemented learning by doing for a robot. The robot maintained a voting matrix, wherein the matrix encoded and stored the effects of every single behavior of the robot with respect to performing the respective task (context). There is a learning signal which is a metric of the correlation between memory and context. If memory and context had a direct correlation, then the robot was invoked to implement the actions learned in the past context and apply them to the present situation. Through this matrix codes the robot learned through experience (picks the codes that prove beneficial).

The addition of new contexts provided the learning gain. This work did not consider task openness.

Fisch et al. (2009) presented their work in a classification domain. The agents were shown items and were asked to classify them. They modeled their agents with some characteristics of learning by doing. They defined learning by doing as the ability of the agent to improve its expertise and skillset through practice and self-perfection. When a new situation came into the environment, the agent created a premise for this new situation and asked a teacher to help with the conclusion of this premise. In this way, when applying already existing knowledge, the agent improved its knowledge with the help of a teacher. The teacher did not support the agent at every step but only with the premises that were presented by the student. Learning gain is a measure of the new conclusions learned by a student from a teacher. This work's environment is stable and there is no uncertainty accounted on cause of openness.

Research work by Jia-hai et al. (2005) demonstrated learning by doing in a negotiation domain. Agents were programmed to negotiate for the participating human parties via the Internet. These agents were modeled to be adaptive to the market situations as well as they learned by the previous actions taken by them and the corresponding rewards achieved, over a period of time. The agents maintained a strength parameter throughout the negotiation. If taking a certain action under a given state proved fruitful then its strength parameter was increased for the next round of negotiation. The initial strength value was set by the agent based on its experiences so far, and would be updated throughout the negotiation process by reinforcement learning. This work accounted for learning gain in the form of a numeric increase in the strength parameter. This learning gain thus is similar to the approach used in our framework as the learning gain in our work is measured as a change of a numeric entity. We account for openness in the environment as well, unlike this work.

Blikstein and Wilensky (2005) presented an agent-based simulation as a powerful learning tool for Material Sciences. The paper reported a user study of a computer-aided learning environment, designed specifically for Material Sciences. They proved that a rich and motivating learning could be achieved through learning by doing, by humans, where the humans learned by carrying out various activities related to Material Sciences. Since this is a user study, there is no algorithm that describes the human learning through observation, just a theoretical perspective of it. The learning gain was the addition of new knowledge achieved by doing. There was no openness modeled in this work.

As seen from Sections 2.3.1, 2.3.2 and 2.3.3, there have been research approaches that equip their agents with learning, while working in a team. However, all these approaches except (Jia-hai et al. 2005) computed their learning gains as an addition of knowledge, context, history of actions, or behavior maps. However, like (Jia-hai et al. 2005), we compute learning gain as a numeric entity.

The agents in our work are modelled on a framework following human-inspired learning strategies Chen (2017). Specifically, we incorporate the **Bandura's Social Cognitive Learning Theory**. This theory states that learning can be achieved by doing and observation. The learning gain curve is a downward shaped parabola for an agent that learns a specific capability under a given skill. This is because when an agent begins with little knowledge, it learns more quickly as it begins to gain expertise. This occurs to a peak point, at which the marginal learning gains begin to slow down.

The learning by observation consists of four levels: *attention, retention, reproduction, and motivation*. For learning by observation, the learner should have skills at par with the teacher to be able to learn from it, and they should also be working on the same task. If a learner learns through observation during a task execution, after successful completion of that task, the learner updates its capability vector based on the gain from observation. Since learning by observation does not cost the agents at all, this keeps the agents constantly motivated to learn.

Thus, as it is seen from Section 2.3, there is research that integrates human-inspired learning approaches in the environment, we summarize three types of such learning: learning by observation, learning by reflection and learning by doing. In our work, we have employed learning by doing and learning by observation Chen et al. (2015). We indeed could further enrich our work by extending the learning by reflection models from Wu et al. (2011) and Schon (1987). We compute learning gain as a numeric entity just as computed by Jia-hai et al. (2005). We could integrate their learning model into our work, since their learning gain computation fits in our work, as is it measured as a numeric entity. The other works measure numeric entity as an addition of knowledge, context, history of actions, or behavior maps. Though this is not something we can employ right away in our work because our learning gains are a numeric entity, but we can look at these types of learning gain computations as a part of our future work to further enhance our model. One major difference between the works specified above and our work is that these works did not account for an uncertain environment, such as an environment which employs openness. We tackle learning in an open environment by employing probabilistic models which is further described in detail in Section 2.4.

2.4 Tradeoff Learning versus task accomplishment

There are research works that employ a tradeoff between different entities such as exploration versus exploitation or agent's personal preferences versus an agreement, etc. when crafting their coalition formation. In our work, we incorporate the tradeoff between learning versus the accomplishment of tasks Chen et al. (2015).

For example, Liemhetcharat and Veloso (2010), did not model learning but introduced a risk factor. Risk factor was a measure of the metric of risk associated with taking up the assigned roles (Roles were allocated based on the capabilities calculated, e.g. in soccer, an attacker, a defender, etc.). This risk factor depended on the capabilities calculated, and the capabilities in return depend on the observations of the teammates. Risk factor was a metric which helped the agents' tradeoff between taking up a role and its own capabilities. But it may take long for a team to have enough observations before to derive the capabilities and in return slowing down the risk calculation and hence role assignment. So, by incorporating learning, the agents can learn the risk factor dynamically instead of having to depend on the capabilities of itself and the teammates. For future work, we can take into account assigning roles to agents and that's when the risk factor modeled in this paper can be used to enrich our work. This work does not take learning or openness into consideration.

In another study, Emery-Montemerlo et al. (2004), proposed an algorithm for dealing with the intractability in Partially Observable Stochastic Games, by converting it into smaller series of Bayesian games. They tested their algorithm in a robot tag game, where the robots worked in teams to chase down a target. They state that their algorithm "Traded off limited look-ahead in uncertainty (no access to the whole world at every timestep) for computational feasibility, and resulted in policies that are locally optimal with respect to the selected heuristic". There was no communication amongst the agents. The agent maintained a belief state based upon its observations (since no communication) of actions in the environment, and it then conditions its policies for the next action selection based on this belief state and the action histories. The Bayesian game approximation maintains a type space for each robot that includes its position as well as its observation at each timestep. The probability of such a type could have been updated both by using the policies computed by the algorithm and through a mutual observation. This is how the agents looked ahead into the immediate round of coalition formation to approximate the reward gain in the future. This work did not consider openness. There is no learning accounted for in this paper and nor is there openness accounted for. For now, we do not need the position of our agents at any point of time, so it is unlikely to extend this work to our work.

Chalkiadakis (2007) presented his work on multiagent coalition formation. He introduced a tradeoff between exploration and exploitation while his agents considered the next action to be taken. The agents addressed the tradeoff between accumulating short-term versus long-term rewards. This tradeoff policy incorporated the tradeoffs between exploration and exploitation, both with respect to the underlying (dynamics and reward) model, and with respect to the behavior of other agents. But what would happen if an agent left the environment or some new agents joined? Will this change the model that the agents have modeled for the behavior of the agents? We could test these notions by introducing openness and hence probabilistic modeling in their environment.

Stone and Kraus (2010) presented a multi-arm bandit problem where the teacher and the learner agent together tried to achieve a common team goal of collecting maximum number of cans. For the teacher, there is a tradeoff between teaching or gaining more rewards for itself. At any point, the teacher can either pull-off its best move and earn rewards for itself or teach the learner. The teacher teaches the learner by pulling arms and displaying the resulting rewards for this action to the learner. For finding the optimal actions for the teacher, the authors had two types of distributions for the arms, first a discrete distribution and second a Gaussian distribution. For discrete distribution of the arms the authors had a polynomial time and memory algorithm, using finite horizon dynamic programming. For the Gaussian case, they convert the instance to a discrete one, and then solve it as a discrete distribution mentioned earlier.

In another research, Shintani and Ito (1998) presented a negotiation algorithm for a distributed meeting scheduler in a multiagent environment. For scheduling a meeting, the agent had to negotiate with the other agents in the environment and reach a mutual decision. For this social decision, this work had to clarify a trade-off between "reaching a consensus" and "reflecting private preferences in the social decision". With a view to improve the trade-off, this work proposed a new multi-agent negotiation protocol that had an effective characteristic function and the persuasion protocol. The characteristic function was used to map the user's (human-user counterpart assigned to the agent) preferences in the negotiation process for the coalition formation of the agents. This work did not employ learning or openness. For our future work, we could integrate this model into our work and include private preferences for agents in the social decision. There is communication amongst our agents presently, so the coalitions are not formed as a part of mutual decision but the auctioneer plays a major role in assigning tasks and hence resulting in a coalition formation. For future work, we could also include the coalition formation as a mutual decision of the agents by getting rid of the auctioneer (assigner) and then we could extend this model to employ a tradeoff between consensus and private preferences.

Coalition formation amongst agents to tackle disasters had been studied by Ramchurn et al. (2010). They implemented a tradeoff between maximizing the working time of a coalition and not losing a future task. To deal with no losing any future tasks, their algorithm performed a one-step look-ahead to find out the consequence of each coalition completing that future task. This will not be fruitful in an open environment since there is no surety of a future task. The one-step look-ahead will then have to be done in probabilistic manner. Our work can help enrich this work and also make their environment more realistic by introducing openness.

In another study, Kenari et al. (2011) studied coalition formation in multiagent systems. They implemented Infinite Horizon technique to define the optimality equations for their Partially Observable Markov Decision Process. They mapped a tradeoff between present rewards and future rewards for the coalition formation. They state that, when the decision for coalition formation is being made, the agent should not solely depend on the immediate reward. The expected discounted future reward should also be considered. The agents can use the transition model (probability of reaching a state by performing an action) to predict the future coalitions that will be formed. But by using the updated beliefs in prediction of the next coalitions, agents' decisions will be more rational and the sequential rationality will be guaranteed. This tradeoff is similar to the way we deploy our tradeoff. We tackle the tradeoff between learning and task accomplishment by using probabilistic modeling.

Khandaker and Soh (2010), designed a web-based collaborative learning tool. Their proposed Multiagent Human Coalition Formation comprised of intelligent agents that firstly, learned a model of the students by keeping a track of their activities, secondly, a Bayesian Network is learned for a probabilistic modeling of groups of agents' models onto their current as well as future task rewards. Using their learned models and this mapping, the agents are now equipped to negotiate to form student groups. This helps the students to perform current as well as future tasks efficiently. This work did not take into consideration openness. We could extend this work by including the Bayesian network for learning a probabilistic modeling of groups of agents' models onto their current as well as future task rewards.

As seen from Section 2.4 there are works that employ tradeoff between private preferences and coalition formation, exploration and exploitation, etc. The tradeoff in this study is employed in a different manner compared to these works though the underling spirit is the same. Neither of these works take into consideration, openness. Openness causes uncertainty in the environment because new tasks and agents entering and old ones leaving. However, our environment integrates openness which causes uncertainty in choosing tasks. We could enrich these approaches by introducing openness and hence extending our probabilistic modeling.

In our work, an agent's objective is to maximize the cumulative reward earned by its user over the sequence of tasks. This requires *non-myopic* planning. However, due to uncertainty caused by agent **openness**, this makes it difficult for the agent to predict which other agents would be present in the environment and what their bids would look like. Thus, the agent needs to estimate an expected task reward that accounts for this uncertainty. Our task selection strategy incorporate **probabilistic modeling** (Chen et al. 2015) *to help the agents deal with the uncertainty caused on account of openness*. This requires the agents to model the likelihood of a given task being assigned, given the observation of the environment. The three probabilities that determine the likelihood of succeeding are 1) Probability of agent winning the bid, 2) Probability of task being assigned, given that the agent won the bid, and 3) Probability of successfully completing the task, given that the task is assigned. Then we compute the expected utility of the revised expertise/skills given tasks, in an open and dynamic environment. *In real-world situations, trading-off learning and task accomplishment needs to consider openness, and this is one of the key contributions of this work.*

2.5 Diagrammatic Summary of the Related Work

We present a diagrammatic summary of all the literature reviews in Figure 2.1. Note, references that did not have Diversity and Openness researched are at the origin of the Figure 2.1.

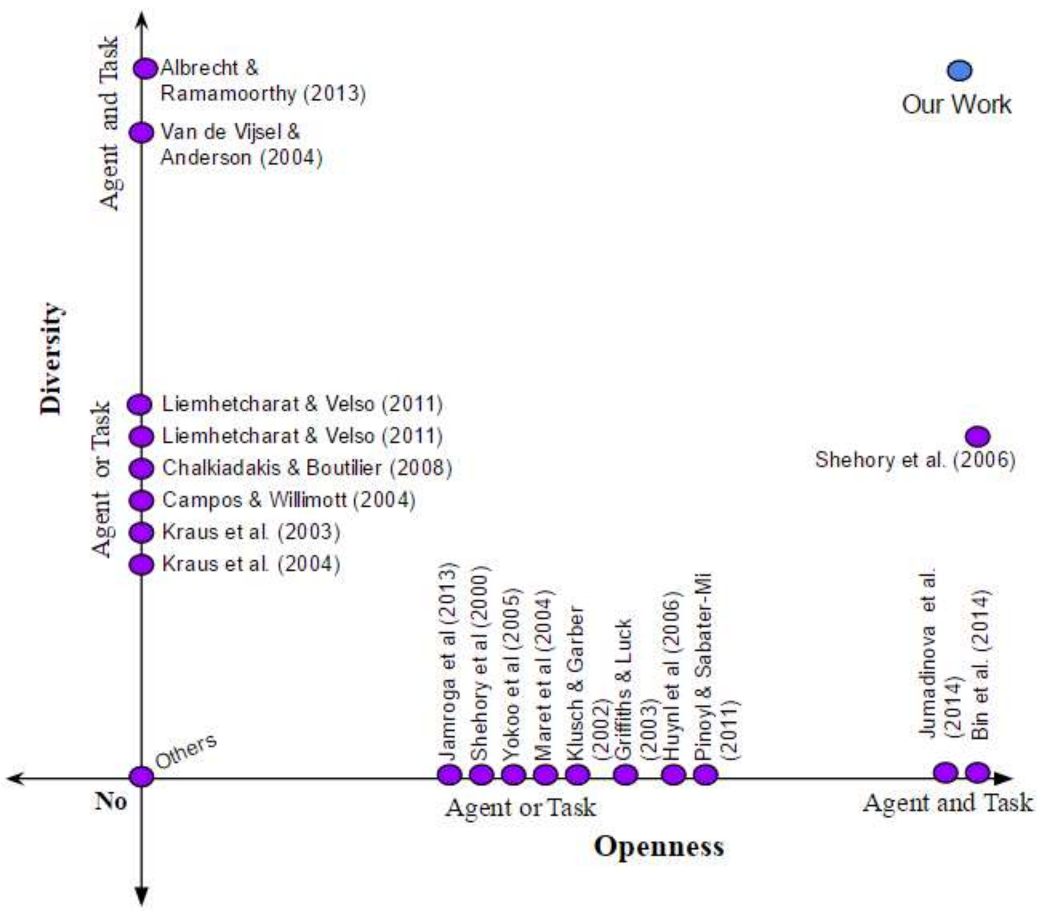


Figure 2.1. Summarization of the Related Work.

Chapter 3

METHODOLOGY

In this chapter, we describe the framework of the model Chen (2017) we extended to include diversity and carried our experiments. We then describe the approach we took to conduct analysis on this model. Following which we present a new approach which will further help enhance the robustness of this model.

3.1 Framework

This section presents the framework of the model Chen et al. (2017), Chen (2017) that we extend to incorporate diversity. The framework helps us simulate our multiagent environment for the problem of agent-based collaborative task assignment.

The environment, comprises of three main **components**:

- The agents present in the environment (A)
- The tasks present in the environment (T)
- The auctioneer used for assigning tasks (CA)

The environment has five main **qualities**:

- Autonomy
- Diversity
- Openness
- Modelling Human Learning

- Probabilistic Modelling

3.1.1 Environment Component Description

This section gives an overview of the three major components that comprise the environment: Agent (Section 3.1.1.1), Task (Section 3.1.1.2), and Auctioneer (Section 3.1.1.3). Agents are the entities that bid for the tasks for their human counterparts. Auctioneer is the central administration responsible for managing the bids submitted by the agents and allocating the tasks to the appropriate agents.

3.1.1.1 Agents

Agents (A) are the software entities that bid for tasks on behalf of their human counterparts. Each agent (a) has a set of capabilities ($\boldsymbol{C}$), where each capability ($c \in \boldsymbol{C}$) has a level of expertise associated with it, $c \in [0,1]$. The level of expertise defines how good an agent is at that capability and it is also the metric that helps the auctioneer decide which agent is the best fit for which task. For example, suppose a software engineer has the skillset of {Coding, Debugging, Writing, Analyzing}. The software engineer will typically have very good level of expertise for Coding and Debugging. However, the engineer—also an agent—may not be good at Writing and Analyzing. These levels of expertise also help the auctioneer determine which agent is a better fit for a given task. For example, a Writer will have a higher level of expertise in Writing as compared to a Software Engineer. Hence, for a task that requires the subtask of Writing, the agent with highest level of expertise for Writing will be picked (provided that agent has bid for that task), over the other agents.

To investigate diversity, we have added *agent types* to the original framework by Chen (2017). The agents are categorized as one of three types: Apprentice, Generalist, or Specialist. The three levels are determined by the level of expertise in capabilities of the agents. An Apprentice is a novice and does not have a level of expertise as good as the other two types. These can be thought of as interns in a company. Interns are not skilled at most of the tasks and are in the process of acquiring knowledge to get better. Generalists are the agents that have a decent level of expertise for a spectrum of capabilities. These can be mapped to Software Developers in the real-world scenario. Software Developers have a decent level of expertise at writing code, debugging, testing, and writing documentation. Thus, Software Developers can be either assigned to various tasks such as Coding, Testing, or Writing Documentation. A Specialist is an agent that has a good level of expertise at fewer capabilities as compared to a Generalist. For example, the Managing Director of a company, even though she might have a set of capabilities like Management, Coding, Writing, and Cooking, she has a very high level of expertise in Management.

The agents also have an ability to learn by observing its teammates and/or performing tasks, and thus improve its capabilities over time. We describe learning in detail in Section 3.1.2. Every time it learns, there is an increase in the level of expertise of its capabilities. This increase in expertise also helps it become a higher agent type, such as promoting itself from an Apprentice to a Generalist. This in return helps an agent be more qualified for bidding for more tasks and increase its chances of being assigned a task. Thus, an agent's capabilities are not static and evolve dynamically with the teammates it encounters or the tasks it performs.

3.1.1.2 Tasks

The tasks (T) in the environment are the entities that the agents bid for. These can be mapped to real-world tasks like, Software Development, Cooking, Writing, Painting, etc. Tasks are composed of subtasks (t). For example, the task of software development will require subtasks like $\{$Coding, Testing, Documenting, Managing$\}$. Each of these tasks requires a specific number of agents ($n_{agents} \in \mathbb{N}$) to accomplish this task. Each subtask within a task also requires a capability, and this capability has a threshold requirement ($qt \in (0,1]$) to be met. For every auction round, we have a fixed number of tasks in the pool for agents to bid on. Each of these tasks has a reward $R(a, T) \in \mathbb{R}$ associated with it, and this reward is only granted to the agents that successfully accomplished this task. Since there are multiple agents that work on a task, the reward is split amongst the agents depending on their share of contribution.

During each round of auction, each agent casts a single bid, if the bid is won, the agent is then allocated to only a single subtask for every auction round. A task is auctioned off if and only if each subtask under the task is successfully assigned to an agent. Thus, there might occur a situation in which an agent might get assigned to a subtask, but the task never gets auctioned off on account of having insufficient number of agents for the other subtasks for that task. If an agent does not find a task suitable for it then it might choose to not bid for that auction round.

3.1.1.3 Auctioneer

This is the central administration in the environment responsible for allocating tasks based on agent bids, and serves more as a convenience for our simulation system. The auctioneer has access to the tasks and the bids in the environment. For every auction round, there is a set fixed number of tasks available in the environment for the agents to bid on. For the agents to have access to these tasks, the auctioneer posts them on a blackboard. Blackboard is a common entity which displays all the tasks available, to which all agents and the auctioneer have access. All the agents get access to the Blackboard simultaneously at the start of every auction round. If an agent wants to cast a bid for a particular subtask, its bidding amount is its own corresponding level of expertise for that subtask. Note that agents do not have any knowledge about the bids of other agents in the environment or the other agents present and their capabilities, as this is part of the requirement of having no or minimal pre-coordination among the agents, as discussed in Chapter 1. Accordingly, neither do the agents have any idea about their perspective teammates. If an agent does not find a suitable task, then it may not bid for that auction round and sit out. At the end of an auction round, the auctioneer has all the bids. It then allocates a subtask to the agent with the highest bid. If all the subtasks within a task are auctioned to an agent, then the task is considered to be auctioned off. The agents that have been assigned to the same task are considered teammates. Once the task is auctioned off, in the current design, it is assumed that the task runs to completion. Each task takes one tick to be completed. Since the task completion is a collaborative task of the teammates and each teammate has a different contribution to the task (depending on the level of expertise), the reward each teammate receives is based on the total reward

(R_{Total}) being split by the share of its contribution, thus keeping it fair for all teammates. The reward for a particular agent a of the n agents that worked for a given task T for every i^{th} subtask, t_i, composing T each requiring a quality threshold qt_i, is calculated as shown in Equation 3.1, from Chen (2017):

$$R(a,T) = \sum_{t_i \in T} \frac{\delta(a,t_i) \cdot qt_i}{n_i \cdot qt_i} R_{Total} \tag{3.1}$$

where $\delta(h, t_k)$ = 1 if agent a was assigned to subtask t_k, else 0.

Our publish-subscribe system Woolridge (2009) for black-board auctioning works as shown in the Algorithm 3.1 Chen (2017).

```
procedure AUCTION
    B ← Blackboard Messages
    for message b in B do
        T ← Task in b
        A ← All agents bids for T
        n_k = number of agents needed for T to be auctioned off
        S ← ∅ //set for assignment of agent to a subtask
        for subtask t_k in T do
            if |A| >= n_k then
                sort(A) //Sort A based on high to low capability
                a_i ← pop last agent from sort(A)
                if cap_{i,k} >= qt_k then //lowest capability agent meets quality threshold requirements
                    for j = 1 : n_k do
                        Add (a_j, t_k) to S
                else
                    Task not auctioned off, no qualified bidders
            else
                Task not auctioned off, insufficient teammates
    Post assignment S
    Remove b from B
```

Algorithm 3.1 Auctioning algorithm from (Chen 2017).

3.1.2 Environment Qualities Description

The three primary functions comprising the coalition formation process include: (1) an agent finding a suitable task, (2) coalition formation and (3) execution of the task. There are environmental factors that make it difficult for an agent to decide which tasks are suitable, and how an agent should decide which tasks are better than the others if there are multiple suitable tasks. There are also issues with coalition formation when agents have to be assigned to a task for a task to be successfully completed. In particular, we see four key challenges: (1) forming coalitions with no or minimal pre-coordination, (2) forming coalitions under open environments, (3) forming coalitions under diverse environments, and (4) forming human coalitions where human learning is present. We have seen how these challenges impact and make coalition formation complex from Chapter Introduction.

3.1.2.1 No or minimal pre-coordination

As identified as one of the key challenges in multiagent coalition formation problem, no or minimal pre-coordination dictates that an agent's knowledge or awareness of the other agents is non-existent or extremely limited at all times. Indeed, the agents in our environment do not have any knowledge about its fellow agents or their capabilities. Unlike most of the research studies in this field where the agents can communicate with each other, our agents do not communicate with each other at all. The only thing known to the agent are their own capability vector and the tasks that are up for bidding for a given auction round. They are not even aware of the bids cast by other agents.

As mentioned earlier in Chapter 1, with minimal or no pre-coordination helps us better map to certain types of real-world problems. For example, in a basketball pickup game scenario, the individual players often do not have the liberty to pick their teammates as they need to form, say, a team of 3 or 5 quickly with whomever is available at the basketball court, could comprise of teammates that have never even met before, let alone played together. However, the team can play together as a whole and also get better with learning from their teammates and/or through the practice sessions.

3.1.2.2 Diversity

We take into consideration diversity in our environment. As described in Chapter Related Work, there are very few works that take into consideration diversity in the environment. However, in the real world, the environment is diverse when it comes to both humans as well as the tasks that they work on.

Agent Diversity. Agent Diversity (AD) accounts for the variety amongst the agents. We classify our agents into either of the three categories: (1) Apprentice, (2) Generalist, and (3) Specialist. All agents in the environment fall under at least one of these three categories. All the agents have the same number of capabilities; however, the distinguishing factor is the level of expertise at which these capabilities are.

Gaussian distributions help us pick the number of capabilities each of the agent types should be skilled at and also the level of expertise for these capabilities. The general form for the distribution of number of capabilities *n* for an agent a is $n_a \sim N(k, \sigma_{count}^2)$, where *k* is the mean and σ_{count} the standard deviation, while the skill level distribution for capability c_i for agent a is $c_{i,a} \sim N(\mu, \sigma_{level}^2)$, μ being the mean and

σ_{level} the standard deviation. Table 3.1 shows the μ and k, or mean values, for the Gaussian distributions from which each user draws its set of capabilities, where $\mu_{low} < \mu_{med} < \mu_{high}$, and $k_{low} < k_{med} < k_{high}$. That is, on average, a specialist has k_{low} number of high-level (μ_{high}) capabilities and k_{high} number of low-level (μ_{low}) capabilities; an apprentice has a high number (k_{high}) of low-level capabilities; and a generalist has a mixture (k_{med}) of low and medium-level (μ_{low}, μ_{med}) capabilities. We define the values of these metrics used for our experimentation in detail in the Chapter Experimentation and Results.

	k_{low}	k_{med}	k_{high}
μ_{low}		Generalist	Apprentice, Specialist
μ_{med}		Generalist	
μ_{high}	Specialist		

***Table* 3.1**. *μ and k values for Agent Capability Set Distributions.*

Apprentices. We define Apprentices as the agents who have low level of expertise for all their capabilities. In the real-world these can be thought of as interns at a company. They are not as skilled as their capabilities yet and still have a lot of learning and improving to do. However, the company does need Apprentices to teach and guide them, and help them become the future of the company. They are the ones to whom the knowledge is passed down to, which is really important when openness also comes into picture, as some skilled worker might just leave the environment abruptly. For example, consider a case where there is only one Software Engineer in the project that knows how to code in Java. Further consider a scenario where this Software Engineer who is really good at coding in Java suddenly leaves the project. This would hamper the project. However, if there were interns or Apprentices in the project then this intern could have been learning Java from the skilled Software Developer, and in case when the Software Developer leaves, the intern can start contributing in Java. Apprentices are also needed to carry out easy tasks which would not be as beneficial if allocated to a professional. Since a professional could instead be assigned to a more difficult task as he or she has high desired skills for that task, unlike an Apprentice. For example, an intern could be hired for writing documentation, or adding comments to the code, at the same time the intern could be learning from professionals by observing them.

Generalists. We defined Generalists as agents that have a decent level of expertise at a wider variety of capabilities. Generalists can be thought of as Software Engineers in a company. These people have fairly good knowledge about a variety of Software Engineering domains like coding, documenting, debugging, testing, etc. Thus, making them eligible to work on a wide variety of tasks. These types of agents are needed in the environment since they are the drive force. Also, they can help the Apprentices evolve by displaying how tasks are performed, for Apprentices to learn by observation.

Specialists. We conceptualize Specialists as agents that are good at few capabilities. These can be mapped to a Machine Learning Engineer within a company. A Machine Learning Engineer can be thought of as a Generalist who has decent Software Engineering capabilities but is really good at the Machine Learning subfield, making her/him Specialist at Machine Learning. Specialists are needed in the environment because these are the ultimate source of knowledge in the environment. They have mastery over a skill which neither an Apprentice or a Generalist has still achieved. Very difficult tasks which require very good knowledge of a given capability can be assigned to Specialists.

Task Diversity. Task Diversity (TD) is used for characterizing the variations in the tasks available in the environment. As seen in the real world, not all tasks are of the same type. Some of them are easy while others are comparatively difficult to solve. Some of them require agents that have really good expertise in capabilities while some do not. Thus, we map the real-world picture of Task Diversity in our model. We split our TD into three categories: (1) Low, (2) Medium, and (3) High. These three categories are conceptualized based on the degree of variety in the tasks available in the task pool. In an environment with c unique capabilities, and all tasks having s subtasks that require those capabilities, the number of possible subtasks combination is $c\ choose\ s$, or C_s^c. We assume s remains constant, and as c becomes greater, so does task diversity level increase. The values of c and s are detailed in the Chapter Experimentation and Results.

3.1.2.3 Openness

This section talks about the environment quality of openness. Our environment is not static: new agents and new tasks can enter and old ones can leave the environment, making the environment open.

Agent Openness. Agent Openness (AO) is conceptualized as the rate at which new agents enter or existing ones leave the environment. For example, in a company, new employees keep joining and old ones keep leaving. Introduction of new employees into the environment signifies potentially new expertise for existing employees learn from. However, department of existing agents could hamper the system, since when an agent leaves the environment so does its level of expertise in his or her capabilities.

Task Openness. Task Openness (TO), correspondingly, the rate at which new tasks enter and existing tasks leave the environment. For example, suppose that the primary service or activities of a company are building websites using Javascript. Then with the release of HTML5, using Javascript for websites has become non-preferable. And now there is a new task at hand: coding websites using HTML5. Another example is the procedure for a disaster response operation. The disaster response consists of a number of stages (Wikipedia, 2017) such as, evacuation, search and rescue, assistance, and restoration.

There are a number of tasks involved under each stage in a disaster response mission. In the beginning, during the search-and-rescue stage, there are tasks such as: searching for survivors and victims, retrieving survivors from debris, applying emergency care to survivors, moving victims to appropriate places, etc. Then, as time progresses, the assistance stage takes hold where food and shelter are provided to the survivors. Thus, tasks involved during the assistance phase are food-related such as collecting, moving, and distributing food and water supplies, and shelter-related such as building tents and providing sanitary facilities. As the disaster response moves from one stage to another, some tasks become obsolete or less significant, while other tasks become more important and necessary. This exemplifies task openness. Furthermore, if a disaster strikes again at the same place—such as aftershocks after a major earthquake, then the cycle could re-start at the search-and-rescue stage, bringing back those tasks carried out during that stage.

3.1.2.4 Human Learning Model

We equip our agents to be able to learn like real-world humans, which helps us maps the real world more accurately. We focus on two types of learning: (1) learning by doing and (2) learning by observation. We extend our learning models from Chen et al. (2017).

Learning by doing. Learning by doing Chen et al. (2017) is learning by performing a given task at hand that leads to an increase in the level of expertise of the capability used while performing that task. Roediger III and Smith (2012); Wifall et al. (2014) show that different tasks have different learning curves. For example, some tasks might require a steep learning curve but once the skills are acquired it is easier to execute these tasks (like learning to play a musical instrument), whereas on the other hand, some tasks might be quick to pick-up but difficult to become an expert at (playing chess, for example). Also, when an agent has a low level of expertise in a given skill, it tends to learn lesser, but as and how its expertise increases its learning rate increases as well. However, this rate increases only up to a certain peak point after which it learns lesser. This makes the learning curve a downward facing parabola. Thus, for an agent a's gain via learning by doing for performing a subtask, with skill *s* with a learning curve capped by α_{do}, using its capability $cap_{a,s}$, we have:

$$\Delta_{do} cap_{a,s} = c\dot{a}p_{a,s} = \alpha_{do} \cdot cap_{a,s} \cdot \left(1 - cap_{a,s}\right) \quad (3.2)$$

Learning by observation. Learning by observation Chen et al. (2017) is based on Bandura's Social Cognitive Theory Bandura (1968), which states that there are four stages involved in learning by observation: (1) Attention, (2) Retention, (3) Initiation, and (4) Motivation. Our model borrows clues from pattern to model the learning by observation. A user pays attention to the performance of its teammates while working on the same task, and in updates its capability after the observation to retain it in memory. Initiating the observed skill can only be possible if the agent is at a comparable level of expertise as the agent it is learning from. The learning gain follows the same sigmoidal curve as for learning by doing. Thus, we model the learning gain function of an agent a observing a teammate b performing subtask τ_l using skill s as follows:

$$\Delta_{obs}cap_{a,s} = \begin{cases} 0 & otherwise \\ \dot{p} & 0 \leq qt_s - cap_{a,s} < \beta \end{cases} \quad (3.3)$$

where β is the threshold under which $qt_s - cap_{a,s}$ is small enough for learning by observation to take place. Note that in Equation 3.3 above, we use $qt_s - cap_{a,s}$ to denote the difference between the capability of the learner agent a (i.e., $cap_{a,s}$) and the capability of the performing agent b (i.e., qt_s), instead of using $cap_{b,s}$. This is because of two reasons. First, using $cap_{b,s}$ would mean that agent a has information or knowledge of the performing agent b's capability. Since we are dealing with no or minimal pre-coordination as discussed earlier in Chapter 1, we aim to constrain the environment such that this information is not shared so readily. Second, when an agent is successfully assigned to use its capability or skill s for a subtask τ_l, even if its skill level is higher than the required quality threshold of the subtask, qt_s, we assume that the agent would only contribute at least that level of skill to complete the subtask. Thus, we use qt_s in Equation 3.3 above.

The learning gain $\dot{p}$ for observational learning is modeled as:

$$\dot{p} = \alpha_{obs} \cdot (qt_s - cap_{a,s}) \cdot \left(\beta - (qt_s - cap_{a,s})\right) \quad (3.4)$$

where α_{obs} refers to the cap for the corresponding learning curve for observational learning for that capability. If the level of capability between the learner agent and the performing agent is too small or too large, then learning by observation is not effective. Here, we use β as the upper bound on this difference: if the difference is greater than $\beta,$ then the gain from learning by observation is zero.

Also, note that in the model, if an agent is already capable of performing a subtask, then the agent does not learn by observing another agent performing that subtask.

3.1.2.5 Probabilistic Modelling

Section 3.1.2.3 reflected upon the openness in the environment introduced on account of agent and task uncertainty. Task openness causes uncertainty in the environment, because the agents are not sure if the task they successfully executed will be available in the environment or not during the next iteration. Even after an agent has won a bid for a given task, the agent cannot be sure if the task will be allocated since it is unsure of the agents that will leave or enter the environment. There can be chances that more qualified agents enter the environment causing the agents to remodel their probabilities of winning a bid, or agents that come together to form teams might leave which might cause the agents left behind to remodel their probabilities of winning bids.

To address these uncertainties, Chen et al. (2015) model the probability that the agent will successfully get assigned task $T \in \mathfrak{T}$ for its user in the current round of bidding as a probability comprising of three parts: (1) the probability that the agent will win a submitted bid, $P_{winBid}(T)$ (i.e., the agent is one of the top n_k bidders for some subtask τ_k), (2) the probability that the task will be auctioned off, $P_{auctioned}(T|winBid)$ (i.e., enough agents bud on the task to form a collaborative team), conditioned on the event that the agent wins the bid, and (3) the probability of successfully completing a task, $P_{success}(T|winBid, auctioned)$, conditioned on it being auctioned off to the agent. Task rewards are composed of two parts: (1) Rewards earned by successfully completing a task, and (2) Gains acquired through learning which will lead to higher rewards in the future. Thus, the total utility for an agent a for a given task T is calculated as the sum of the rewards earned and the gains in the utility from learning:

$$U(a,T) = R(a,T) + U_{Learn}(a,T) \quad (3.5)$$

Thus, by introducing openness in Equation 3.5, the expected utility under uncertainty is calculated as:

$$E[U(a,T)] = P_{winBid}(T) \cdot P_{auctioned}(T|winBid) \cdot P_{success}(T|winBid, auctioned) \cdot \left(E\left[\hat{R}(a,T)\right] + E[U_{Learn}(a,T)]\right) \quad (3.6)$$

The probabilities in Equation 3.6 are updated throughout the presence of the agent in the environment. They are updated based on the experience an agent has with the tasks and teammates in the environment for the given auction rounds.

To learn $P_{winBid}(T)$, the agent takes into consideration its history of bids cast. If an agent is on a winning streak for a given task, then it understands that it is one of the strongest bidders for the given task, thus making it likely that it will win a bid on the task T as well. On the other hand, if an agent happens to lose a bid quite often, then it learns that it is not wise for it to continue to bid for that task as the probability will be low, rendering the expected utility low.

Based on this intuition, the agent considers the s-most similar tasks $S(T)$ that it previously bid on (where task similarity is calculated using the Euclidian distance between the qt_k and n_k values required for the subtasks $\tau_k \in T$). Within these s tasks, it considers the proportion of won bids:

$$P_{winBid}(T) = \frac{1}{|S(T)|+\epsilon'_{winBid}} \sum_{T' \in S(T)} wonBid(T') + \epsilon_{winBid} \quad (3.7)$$

where, $wonBid(T)$ signifies if the bid was won (1) or not (0), ϵ_{winBid} and ϵ'_{winBid} are, small constants providing a non-zero (albeit small) probability of winning a bid, even if the agent has never previously won a similar task (noting that its situation might have changed due to human learning and agent openness).

To learn $P_{auctioned}(T|winBid)$ a similar approach to the above is followed wherein, the similar tasks for which an agent won a bid and also the task was auctioned off as there were sufficient number of teammates present to accomplish the task is calculated as:

$$P_{auctioned}(T|winBid) = \frac{1}{|S(T)| + \epsilon'_{auctioned}} \sum_{T' \in S(T)} auctionedOff(T') + \epsilon_{auctioned} \quad (3.8)$$

where, $auctionedOff(T)$ signifies if the task T is auctioned off (1) or not (0), $\epsilon_{auctioned}$ and $\epsilon'_{auctioned}$ are small constants providing a non-zero (albeit small) probability of the task being auctioned.

To learn $P_{success}(T|winBid, auctioned)$, a similar approach to the above is followed wherein, the similar tasks for which an agent won a bid and the task that got auctioned off and successfully completed as well are taken into consideration. $P_{success}(T|winBid, auctioned)$ is calculated as:

$$P_{success}(T|winBid, auctioned) = \frac{1}{|S(T)| + \epsilon'_{success}} \sum_{T' \in S(T)} succeed(T') + \epsilon_{success} \quad (3.9)$$

where, $succeed(T)$ signifies if the task T was successfully completed (1) or not (0), $\epsilon_{success}$ and $\epsilon'_{success}$ are small constants providing a non-zero (albeit small) probability of the task being successfully completed.

3.2 Approach to analysis

The above framework is used to set up the environment for our investigation stage. We carried out a series of experimentations with different permutations and combinations of

Agent and Task Diversity, and Agent and Task Openness and studied their impacts on the environment as well as the agent reasoning and performance. We took detailed investigations into the results at regular intervals of the total time period the experimentation was run for. This helped us study the environment and agent activities at finer details and hence notice the trends which wouldn't have been visible if the results had only been viewed from a higher level. We investigated the impacts of diversity and openness on a number of factors like the tasks auctioned off, the learning gains, the types of teams and teammates, etc. which is described in great details in the following chapter, Chapter Experimentation and Analysis.

Having examined the results and the trends at finer levels helped us realize that Equation 3.8 could be enhanced further to be able to tackle diversity. This has ultimately led us to proposing a new solution which is presented in Section 3.3 and the results of this new solution are described in detail in Chapter 5.

3.3 Tackling Diversity with New Solution

Equation 3.8 is designed in such a way that it allows an agent to learn not to bid for tasks that did not get auctioned off despite the agent winning the bid. This can prove to be a shortcoming in an agent- and task-diverse environment. Since in a task- and agent-diverse environment, there is a variety of tasks and agents present. Because of this variety of tasks and agents, agents are split across by bidding for different tasks, instead of coming together to work on common tasks. Thus, most of the agents win the bids they cast but do not get the tasks assigned on account of insufficient number of teammates targeting the same tasks. But Equation 3.8 teaches these agents to stop bidding for tasks for which they won bids but never got auctioned off. Thus, most of the agents abandon those bids and look for new tasks to bid on. Diversity causes many possible permutations for the agents to pick the next task, which will then cause them to have insufficient number of teammates. In effect, this causes the agents to keep *chasing* each other and in return reduces the number of tasks that are auctioned off. A reduction of number of tasks auctioned off also impacts the learning gains of agents since they do not work on tasks and hence do not learn by doing and observing their teammates.

Thus, to make Equation 3.8 suitable for both open and diverse environments, we add an additional entity to this equation; the summation of the fraction of teammates that won their bid for the given task, irrespective of whether the task was auctioned off or not. Note, if the task was auctioned off then the value of the fraction is 1. Therefore, we modify Equation 3.8 as shown in Equation 3.10 and we now term this equation as $P_{auctioned+}(T|winBid)$ since it is an enhancement of $P_{auctioned}(T|winBid)$:

$$P_{auctioned+}(T|winBid) = \frac{1}{2*|S(T)|+\epsilon'_{auctioned}} \sum_{T' \in S(T)} (auctionedOff(T') + teammateRatio(T')) + \epsilon_{auctioned} \quad (3.10)$$

where, $teammateRatio(T')$ signifies the ratio of the total number of subtasks that were successfully assigned—for which an agent successfully won its bid—to the total number of subtasks required for the task T. It is a metric which shows how close a team was to filling in all the subtasks and getting a task auctioned off. The higher the value of this metric, the higher the number of subtasks is successfully assigned. Note, in the worst-case scenario, all the similar tasks will not be auctioned off because of only one (or zero) agent bidding on the task, making the probability ~0. The best-case scenario is when all the similar tasks will be auctioned off because all the teammates were present, making the probability 1.

Thus, by adding this new metric, we now modify Equation 3.8, such that it does not just allow agents to learn to start bidding for different tasks—when an agent is able to win its bids but the tasks are not getting auctioned off—but also takes into consideration how close a task was getting auctioned off, in terms of finding sufficient number of teammates to fulfill all its required subtasks. If the task was close to getting sufficient number of teammates to get the task auctioned off then the probability of having the task auctioned off should be higher than one that was missing a large number of teammates. This should lead to agents being more persistent in trying those "close" misses instead of abandoning them. Conversely, if there happens to be a task that had many unassigned subtasks, then Equation 3.10 should guide the agents to bid for different tasks. Thus, instead of chasing each other around, Equation 3.10 could now help channel stray agents to work on common tasks, thus addressing the impacts of diversity in open environments. As shown in Chen et al. (2015), Equation 3.8 can be effective in open environments, and since Equation 3.10 is an extension of it, this new approach should still be fit to work in open environments.

Chapter 4

EXPERIMENTATION AND INVESTIGATIONS

In this chapter, we present the results of the experiments carried out with the objective of investigating the impact of openness and diversity on team formation and how agents learn to form teams over time. As seen from the Chapter Introduction, team formation is a complex task made even more challenging in an open and diverse environment. With a view of analyzing these impacts, our investigations are based on data collected from a set of simulation runs using the multiagent system framework described in Chapter 3.

For our simulation runs, as shown later in Section 4.1, we used multiple configurations in terms of Agent Diversity (AD), Task Diversity (TD), Agent Openness (AO), and Task Openness (TO). For each of these parameters, we used multiple values. For example, AO was set to 0, 0.05, and 0.1 in different configurations. Thus, each configuration is a unique combination of these parameters. For each configuration, we conducted multiple runs (e.g., 100) to obtain the average[2] performance, which included the average number of tasks completed, the average rewards per tick, the average rewards per task, the average learning gains per tick, and the average learning gains per task. For each run, there were multiple agents and tasks in the environment for some duration measured in ticks (e.g., 1000 ticks). We collected data for each agent during each tick.

[2] Throughout this chapter, average refers to the average over all the 100 runs for each of the configurations.

We also collected data on tasks—and their subtasks—and teams. Details of these are provided later in Chapter Implmentation.

In this Chapter, Section 4.1 gives a detailed overview of our experiment setup. Section 4.2 analyzes how diversity and openness impact agent performance. To do so we take a look at the average rewards per tick—the most important performance metric as agents were designed to maximize expected utility or rewards over time, as discussed earlier in Chapter 3—and the impacts of diversity and openness on this metric. We also establish the relationships between the average rewards per tick and the average number of tasks auctioned off (i.e., the number of tasks completed) such that we focus on the average number of tasks auctioned off to subsequently investigate how the impacts come about. Therefore, the rest of the chapter—Sections 4.3, 4.4, and 4.5—investigates the impacts of diversity and openness on team formation and how agents learn to form teams, with respect to the number of tasks auctioned off.

4.1 Experiment Setup

This section gives an overview of our experimentation environment setting. We carried out our experiments under the following environment configuration shown in Table 4.1. Note that we denote each AD configuration as a triplet of agent type percentages in the environment, represented as G-S-A percentages. For example, 25-25-50 signifies that there are 25% Generalists, 25% Specialists, and 50% Apprentices in the environment. In our investigation on the impacts of agent diversity as discussed in Section 4.2 later, we used the following AD configurations: 25-25-50, 0-50-50 (no Generalists, but 50%

Specialist and 50% Apprentices), and 50-0-50 (no Specialists, but 50% Generalists and 50% Apprentices). For these configurations (recall, the metrics defined in Chapter Methodology), we define $\mu_{low} = 0.15$, $\mu_{med} = 0.5$, and $\mu_{high} = 0.85$ for the sampling of capability skill levels. $k_{low} = 2$, $k_{med} = 6$, and $k_{high} = 10$ for the sampling of number of agent capabilities. We use $\sigma_{level} = 0.175$ for sampling capability levels and $\sigma_{count} = 1$ for sampling number of agent capabilities to ensure little overlap between each agent types' distributions.

We took into consideration three levels of task diversity (TD) for our experimentation, High, Medium, and Low. We calculated each of the three levels of TD as shown in Equation 4.1 (Recall the metrics were defined in Chapter Methodology):

$$Low = C_5^9 \quad (a)$$

$$Medium = C_5^{10} \quad (b)$$

$$High = C_5^{11} \quad (c) \qquad (4.1)$$

In the above, C_k^n means *n* choose *k*. Thus, in the TD = High configuration, the tasks have the highest diversity as each task's subtasks are randomly chosen from a pool of 11 subtasks, for example.

In the following sections, one of the performance metrics that we will keep track of is the number of teams formed during each tick or iteration of the simulation. We use this term interchangeably with *the percentage of tasks auctioned off* or *the percentage of tasks completed* as teams are formed when every subtask under a given task is auctioned off to an agent. An agent can be assigned to only one task per auction round to do one subtask.

Configuration Parameters	Values
# Agents (Total population of agents)	100
# Ticks (#Iterations of the simulation)	1000
Agent Openness (AO)	0.0, 0.05, 0.1
Task Openness (TO)	0.0, 0.05, 0.1
Task Diversity (TD)	Low, Medium, High
Agent Diversity (AD)	%Generalists-%Specialists-%Apprentices: 25-25-50, 00-50-50 and 50-00-50
Agent type	Agent type 1: Generalist (G), Agent type 2: Specialist (S), Agent type 3: Apprentice (A)
Number of Runs per Configuration	100

Table 4.1. Configuration parameters of the environment in our simulations.

4.2 Impacts of diversity and openness on agent performance

This section helps us analyze the impacts of diversity and openness on the rewards earned per tick. Table 4.2 presents the average rewards earned per tick over the 100 runs for each of the AD = 25-25-50, 00-50-50, and 50-00-50, TD = Low, Medium, and High, and AO = TO = 0, 0.05, and 0.1, respectively.

		AD = 25-25-50			**AD = 00-50-50**			**AD = 50-00-50**		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	0.0035	0.0046	0.0070	0.0014	0.0025	0.0051	0.0024	0.0041	0.0065
	0.05, 0.05	0.0066	0.0067	0.0094	0.0029	0.0038	0.0069	0.0046	0.0053	0.0074
	0.1, 0.1	0.0068	0.0076	0.0095	0.0037	0.0056	0.0074	0.0049	0.0070	0.0086

Table 4.2. Average rewards per tick (100 runs for each configuration) for AD = 25-25-50, 00-50-50, and 50-00-50, TD = Low, Medium (Med), and High, and AO = TO = 0, 0.05, and 0.1, respectively, standard error = 0.0004.

Diversity. As seen from Table 4.2 we see that the average rewards per tick is the highest for the 25-25-50 AD configuration, followed by 50-00-50 and then 00-50-50, for all three TD configurations. The 25-25-50 AD configuration performed the best since the diversity in agent types helped the agents learn a more varied set of capabilities at different levels of expertise and work on a wider variety of tasks. As discussed in Chapter Methodology, we referred to Bandura's theory for calculating learning by observation. Based on Equation (3.3), we see that if the difference between an observer's expertise level and a performer's is too large or too small, then the observer does not benefit much from learning by observation. Thus, it is easier for an Apprentice to learn from a Generalist than it is for an Apprentice to learn from a Specialist. Thus the 50-00-50 (i.e., only Generalists and Apprentices) configuration did better than the 00-50-50 (i.e., only Specialists and Apprentices) configuration. It can also be observed that the average rewards earned per tick increased with the increase in the level of task diversity. This happened because, if the level of task diversity is low and agents do not find a suitable task then it is likely going to be difficult for them to find a task through the auction rounds. On the other hand, a high level of task diversity affords agents more opportunities to find suitable tasks and also get better at a wider variety of capabilities. Thus, we see that diversity—agent or task—does have an impact on agent performance. The higher the diversity is in an environment, in general, the higher is the average reward per tick earned.

Openness. Agents in the most open environment earned the highest rewards (AO, TO = 0.1, 0.1) followed by the comparatively less open environment (AO, TO = 0.05, 0.05) and finally the environment with no openness (AO, TO = 0, 0). Openness presents agents with the opportunity to be able to work with new agents and in return be able to learn a wider variety of capabilities at different levels of expertise. Openness also helps agents in finding new tasks that they are good at, which also provides more opportunities for them to sharpen their capabilities. On the other hand, in an environment with no openness, there are no new agents entering or no new tasks entering the environment, which restricts the agents to the same set of potential teammates and the same pool of tasks to consider. This restriction takes away the opportunity from the agents to get better at a wider variety of capabilities. Thus, openness also has an impact on agent performance. The average rewards earned per tick increases in general with the level of openness in our setup. This confirms the findings reported in Chen (2017).

Interactions between Openness and Diversity. When we looked at the average rewards per tick for AO = TO = 0, for AD = 25-25-50, 00-50-50, and 50-00-50, and across TD = Low, Medium, and High, we realized that the rates of rise of the average rewards per tick were much closer between Low and Medium TD (i.e., 0.0011, 0.0011, and 0.0017, across the three AD configurations, respectively.) and were farther apart between Medium and High TD (i.e., 0.0024, 0.0026, and 0.0024, across the three AD configurations, respectively). This pattern was consistent across all three AD and TD configurations in the absence of openness. Thus, there is a pseudo-linear relationship between AD and TD configurations and average rewards per tick particularly in the presence of diversity but without openness. However, the rates of rise of the average rewards earned per tick did not follow a linear pattern when openness was introduced with diversity. From Table 4.2, again, we see that the average rewards per tick across the increasing levels of task diversity for AD = 25-25-50, for AO = TO = 0.05, and 0.1 did not follow a consistent linear pattern. For example, the rates of rise of the average rewards per tick for AD 25-25-50, with AO = TO = 0.05 and 0.1, between Low and Medium were 0.0001 and 0.0008, respectively, and that between Medium and High were 0.0027 and 0.0019, respectively. This shows that the impact caused by openness or diversity individually becomes less predictable when both together are present in the environment. This also helps support our claims made in Chapters 1 and 3 that openness when combined with diversity makes the environment more challenging for agents to reason with.

Since our primary objective is to analyze the impacts of diversity and openness on the team formation and how agents learn to form teams, we are particularly interested in task-related metrics. For example, we look at the task assignments, the team makeup, and the type of agents that come together to form teams. To be able to analyze these task-related metrics, we thus prefer to focus on investigating the tasks auctioned off. But, in order to be able to measure impacts on team formation we need to focus on the most important performance metric, which is the average rewards per tick as discussed above, since the agent reasoning was designed to maximize the expected utility over time. Thus, here we seek to establish the relationship between the average rewards per tick and the average number (or percentage) of tasks auctioned off. A positive correlation between these two metrics will help us use task-related metrics for our investigations that follow. With this in view, Tables 4.3, 4.4, and 4.5 present the correlation between the average rewards per tick and the average percentage of tasks auctioned off over time for AO = TO

= 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, Medium, and High, respectively.

TD = Low	AD = 25-25-50	AD = 00-50-50	AD = 50-00-50
(AO, TO) = (0, 0)	0.8940	0.8875	0.8843
(AO, TO) = (0.05, 0.05)	0.8947	0.8674	0.8855
(AO, TO) = (0.1, 0.1)	0.9582	0.9124	0.9236

Table 4.3. Correlation between average rewards per tick and average percentage of tasks auctioned off for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, respectively.

TD = Medium	AD = 25-25-50	AD = 00-50-50	AD = 50-00-50
(AO, TO) = (0, 0)	0.8974	0.8692	0.8867
(AO, TO) = (0.05, 0.05)	0.9240	0.9339	0.9246
(AO, TO) = (0.1, 0.1)	0.9874	0.9748	0.9853

Table 4.4. Correlation between average rewards per tick and average percentage of tasks auctioned off for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Medium, respectively.

TD = High	AD = 25-25-50	AD = 00-50-50	AD = 50-00-50
(AO, TO) = (0, 0)	0.9238	0.8874	0.8768
(AO, TO) = (0.05, 0.05)	0.9785	0.9857	0.9785
(AO, TO) = (0.1, 0.1)	0.9890	0.9886	0.9839

Table 4.5. Correlation between average rewards per tick and average percentage of tasks auctioned off for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = High, respectively.

As observed from Tables 4.3, 4.4, and 4.5, we realize that the correlation between average rewards per tick and the average percentage of tasks auctioned off is high (all above 0.86). Given this high correlation and the observations of the impacts of diversity and openness on agent performance, for the rest of the chapter, to simplify our discussions, we will concentrate on the task-related metrics for our further investigations for analyzing the impacts of diversity and openness on the team formation and how the agents learn to form teams over time.

Now, to investigate the impacts of diversity and openness, we have the following three investigations (depicted in Figure 4.1) with respect to the task-related metrics:

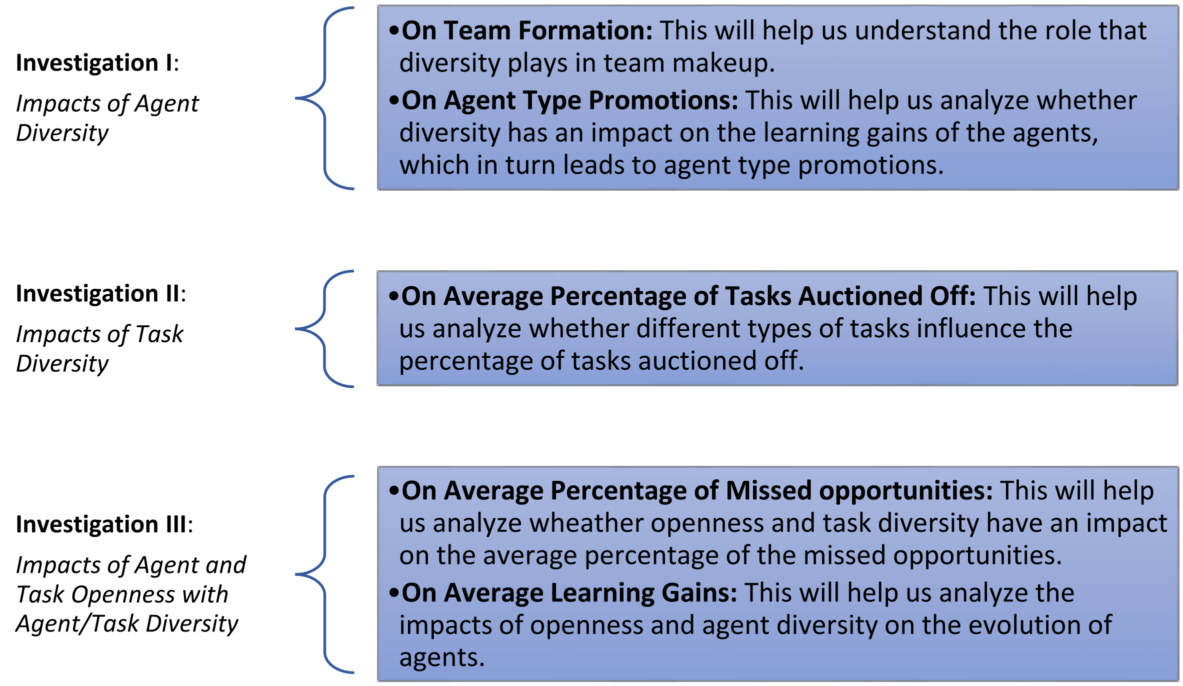


Figure 4.1. Objectives of the three investigations carried out to analyze the impacts of diversity and openness on team formation.

Investigation I: Impacts of Agent Diversity. The primary purpose of this investigation is to help us analyze the impacts agent diversity has on the team formation, with respect to the number of tasks completed (or auctioned off). To help us understand how diversity plays a role, we further look at the team makeup—including homogeneous and heterogeneous teams—and at how agent types are promoted, say, from Apprentice to Generalist, from Generalist to Specialist, as presented in Section 4.3.2.1. Note that type promotion is an indication that an agent learned and became better at its capabilities. This allows us to look at an agent's potential teammates and competitors.

Investigation II: Impacts of Task Diversity. The primary purpose of this investigation is to help us analyze the impacts of task diversity has on the team formation, with respect to the number of tasks completed (or auctioned off). To help us understand how diversity plays a role, we further look at the average percentage of tasks auctioned off under the configurations with different types of tasks present in the environment.

Investigation III: Impacts of Combining Openness with Agent/Task Diversity. Note that Chen (2017) already established that both agent and task openness have an impact on team formation and how agents learn to form teams. Thus, here in this Investigation, we did not treat agent and task openness separately. Further, as argued in Chapters 1 and 3, we suspected that there might be complications brought on by combining openness with diversity. Thus, we put forth here an initial step towards understanding the impacts of combining openness and diversity, by focusing only on one diversity at a time. The primary purpose of this investigation is to help us analyze the impacts openness and task diversity has on the team formation, with respect to the average percentage of missed opportunities. We also take a look at the learning gains under openness and agent diversity to analyze the evolution of the agents.

4.3 Investigation 1: Investigating the Impacts of Agent Diversity

In this section, we focus on the impacts of agent diversity on team formation and the team makeup (Section 4.3.1), and on the impacts of learning and agent diversity (Section 4.3.2). Note, that in all of the following experiments in this Section, we used AD configurations 25-25-50, 50-0-50, and 0-50-50, and set TD = Medium, and AO = TO = 0. We set openness to zero so as to be able to focus on agent diversity completely, this is also the reason why the level of task diversity is set to Medium only.

4.3.1 Teammates and Impact of Agent Diversity on Team Formation

First, we investigate the impact of diversity on the average number of tasks completed (i.e., the average number of tasks auctioned off). This will help us understand how the teams were formed, thus helping us understand how diversity plays a role in the agent types that come together to form teams. We also want to investigate the impacts of a diversified team on learning and in return on team formations.

4.3.1.1 Teammate Statistics and Agent Diversity

This Section presents the results of the statistics of the various teams formed and the tasks allocated under each of the three AD configurations as shown earlier in Table 4.1.

Overall Statistics. Table 4.6 presents the average number of tasks auctioned off for each AD configuration: 25-25-50, 00-50-50, and 50-00-50, with TD = Medium and no openness. Note, average throughout this chapter is the average over the 100 runs for each of the configuration. As seen from Table 4.6, the highest average number of teams (i.e., average number of tasks auctioned off) is formed when all the three agent types (G, S and A) are present in the environment, i.e., in the 25-25-50 configuration. Specialists have certain expertise at which they are better as compared to the other two agent types. For tasks at which Specialists do not have the qualifying expertise a Generalist may step up to win its bid for it. When most of the Generalists and Specialists are allocated to tasks that benefit them the most, the remaining subtasks could be allocated to the Apprentices (if they qualify). These Apprentices could learn from a diverse agent set and improve their expertise through learning, in return proving to be more beneficial to the environment.

	Agent Diversity Configurations		
	25-25-50	00-50-50	50-00-50
# of Tasks Available	1,000,000	1,000,000	1,000,000
# of Tasks Not Auctioned Off (%)	392,101.7650 (39.21%)	496,107.9820 (49.61%)	417,022.6000 (41.70%)
# of Tasks Auctioned off (%)	607,799.5654 (60.78%)	503,809.1183 (50.38%)	582,912.0070 (58.29%)

Table 4.6. Average statistics about the tasks for each AD configuration (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, standard errors = 0.0005.

The average number of teams formed (i.e., average number of tasks auctioned off) in the 00-50-50 and 50-00-50 configurations, where either Generalists or Specialists are absent, is lower than that in the 25-25-50 configuration. Also, it can be seen that the average number of teams formed in 00-50-50 is lower than that in 50-00-50. This is because of two reasons. First, as discussed in Chapter Methodology, we referred to Bandura's theory for calculating learning by observation. Based on Equation (3.3), we see that if the difference between an observer's expertise level and a performer's is too large or too small, then the observer does not benefit much from learning by observation. Thus, it is easier for an Apprentice to learn from a Generalist as compared to an Apprentice learning from a Specialist. Second, though a Specialist has higher level of expertise than a Generalist in a particular capability, a Generalist is qualified or fit for a wider variety of capabilities[3] compared to a Specialist. Thus, when an Apprentice is

[3] Note that the concept of "wide variety of capabilities" means that an agent has a large number of capabilities with non-zero and decently high level of expertise.

teamed up with other Generalists for a task, it will have a higher chance of learning by observation more diverse capabilities.

To further understand the impact on how agents, form teams to solve tasks, let us look at also how they bid for tasks. First, we will refer to the phenomenon of "bidding for a wider variety of tasks" further. This means that the agent bids for different tasks over time rather than bidding for the same task repeatedly. That bidding for a wider variety of tasks is possible because of two situations: (1) the agent learns and expands its number of capabilities of non-zero level of expertise, and (2) the agent learns and increases its level of expertise in some of its capabilities. These situations allow an agent to model and realize over time that its chance of winning a bid for a task has increased or reduced. This drives the agent's selection of the best task to bid for at each iteration. Now, on account of being fit (or qualified to perform) at a wider variety of capabilities, a Generalist can bid for a wider variety of tasks as compared to a Specialist. This in return increases the chance of a Generalist being considered for a wider variety of tasks as compared to a Specialist. Indeed, upon further investigation, we found that Generalists bid for a wider variety of tasks 62% of the times, as compared to Specialists bidding for a wider variety of tasks 57% of the times. In summary, this helps Apprentices become better at a wider set of capabilities—indeed, 98% of Apprentices showed that they had a wider capability set at the end of the simulation—and in return start bidding for a wider spectrum of tasks, leading to more tasks being auctioned off overall in the system.

Team Makeup. Here we investigate further into which agent types came together to form teams. Table 4.7 presents the average number and the average percentage of teams formed by different agent types, under all three AD configurations (25-25-50, 00-50-50, and 50-00-50). To help with our analysis, we also identify homogeneous teams where there was only one type of agents on a team, and heterogeneous teams where there were two or more agent types present.

Total # of teams (%)	**25-25-50**	**00-50-50**	**50-00-50**
G only	58,104.6940 (9.56%)	0	224,592.8510 (38.53%)
S only	53,304.0000 (8.77%)	164,877.0004 (32.72%)	0
A only	48,441.8302 (7.97%)	99,924.1769 (19.83%)	72,222.2888 (12.39%)
Total # of homogeneous teams	159,85.5242 (26.27%)	264,801.1770 (52.56%)	296,815.1400 (50.94%)
G and S only	153,408.1000 (25.24%)	0	0
G and A only	47,165.5102 (7.76%)	0	286,097.0192 (49.08%)
S and A only	29,478.2262 (4.85%)	239,055.3243 (47.44%)	0
G, S and A only	217,897.4290 (35.85%)	0	0
Total # of heterogeneous teams	447,949.2650 (73.72%)	239,055.3243 (47.44%)	286,097 (49.08%)

Table 4.7 Average (over the 100 runs) Count and percentage of the teams formed by the different Agent types for each AD configuration (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, standard error = 0.0004.

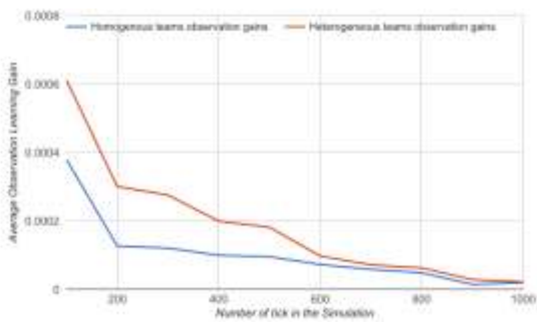
As seen from Table 4.7 it is interesting to see how Apprentices, with very low level of expertise in their capabilities, especially in the beginning in the simulation, still manage to complete tasks on their own (i.e., under "A only"). Upon further investigation, we realize that all of the tasks that only had Apprentices as teammates had a low expertise level requirement and hence all the bidders were of the type Apprentice, whereas the Generalists and/or Specialists available in the environment had submitted their bids for tasks that required level of expertise suiting their skillset. Note that the level of Task Diversity (TD) here was set at Medium and the AO and TO are both set to 0. With the availability of tasks requiring capabilities at different levels of expertise, all agent types had a chance to bid, win, perform tasks, and learn by doing and observation and improve further.

We also see that Apprentices also tried to compete with Generalists and Specialists for tasks. For example, in the 25-25-50 AD configuration, out of all the teams that were formed with only Generalists and Specialists (total of 153,408.1000 average teams, shown in Table 4.7), 52% of these successfully auctioned off tasks actually also received bids from Apprentices[4]. However, the Apprentice bidders were not allocated any of the subtasks because of their lower expertise levels as compared to the Generalist and Specialist bidders'. Thus, even though an Apprentice qualified for a certain subtask, there was a Generalist or Specialist that was more qualified and ended up winning the bid in these situations.

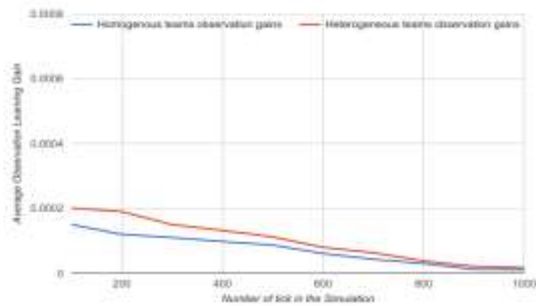
One interesting finding from Table 4.7 is that, for the AD 25-25-50 configuration, the average number of heterogeneous teams was *more* than the average number of homogeneous teams. One reason for this is that the learning gain is higher in a heterogeneous team as compared to a homogeneous team. Since a heterogeneous team has high diversity in the capabilities and levels of expertise in its members, one can learn from multiple types of agents through observation as compared to when in a homogeneous team. To further investigate this, Figure 4.2(a) shows the average learning by observation gain for the homogeneous versus the heterogeneous teams, for AD 25-25-

[4] Note that an agent should be able to learn and model the probability of winning a bid for a particular task as time progresses and thus avoid bidding for a task that it cannot win with its bid. However, in this case where the Apprentices submitted these bids and competed with other, more-capable agents, the Apprentices did so as these were the only tasks that they were qualified for.

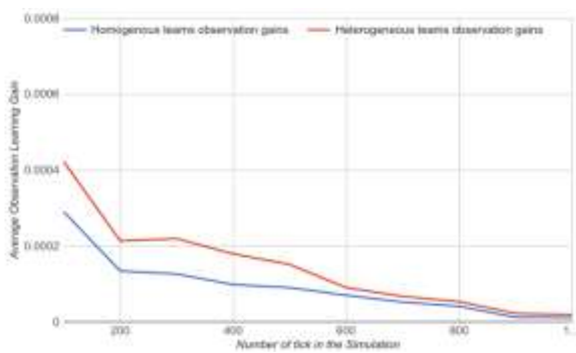
50. As seen from Figure 4.2(a) the average learning gain for heterogeneous teams was higher than that for homogeneous teams. Thus, it is also possible for agents to bid for tasks that require more diverse capabilities—involving heterogeneous members—in order to reap benefits from learning, and thereby forming more heterogeneous teams over time. We see similar trends from Figure 4.2(b) and Figure 4.2(c) for AD configurations 00-50-50 and 50-00-50 respectively.



(a) AD 25-25-50



(b) AD 00-50-50



(c) AD 50-00-50

Figure 4.2. Average gains from learning by observation for homogeneous and heterogeneous teams for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

Based on the above understanding, one would expect that Generalists should dominate the teams as a member in the number of tasks solved. To reassure us of this expectation, we look at the average dominant agent types. An agent type is dominant in a heterogeneous team if there are more members of that agent type than the others combined in the team. Table 4.8 presents the average number of times a given agent type is dominant in a heterogeneous team. Note that we exclude homogeneous teams from Table 4.8. As seen from Table 4.8, Generalists are the highest average number of dominants. This is a really re-assuring finding. Most of the subtasks within a task are taken up by Generalists, since they are the ones having a better skillset along with a decent level of expertise in this skillset, at a wider variety of tasks compared to an Apprentice and a Specialist.

# teams under dominant agent types (% of these teams)	**25-25-50**	**00-50-50**	**50-00-50**
G	372,839.1926 (49.18%)	0 (0)	173,736.0037 (45.32%)
S	137,162.0006 (18.09%)	72,362.2273 (32.01%)	0 (0)
A	48,759.1473 (6.43%)	34,881.1545 (15.43%)	14,354.2033 (3.74%)

Table 4.8. Average (over the 100 runs) number of teams formed by corresponding dominant agent type for each AD configuration (25-25-50, and 00-50-50, and 50-00-50, with TD = Medium, and AO = TO = 0, standard error = 0.0004.

4.3.1.2 Teammate Statistics over Time and Agent Diversity

This Section presents teammate statistics over the entire simulation duration to allow us to observe the *evolution* or *dynamics* of how different agent types formed teams. Thus, in contrast of the *overall* statistics presented in Section 4.3.1.1, here we look at the results after each 100-tick interval. Here, we first take a look at the task related statistics over time. And then we analyze the type of agents that come together to form teams over time.

Overall Statistics: Tasks Auctioned Off over Time. This section presents the task-related statistics for all three AD configurations, at a finer level. By observing the tasks at every 100-tick interval we can analyze how the average number of tasks auctioned off changes over time. To simplify our presentation, here we focus only on the average *percentage* of tasks auctioned off. Figure 4.3 presents the average percentage of tasks auctioned off for each of the three AD configurations per 100th tick. We can see that the average percentage of tasks auctioned off increased over time for all three AD configurations. This is because that, over time, the agents by learning acquired the level of expertise for the tasks available in the environment, and became capable of being assigned more tasks.

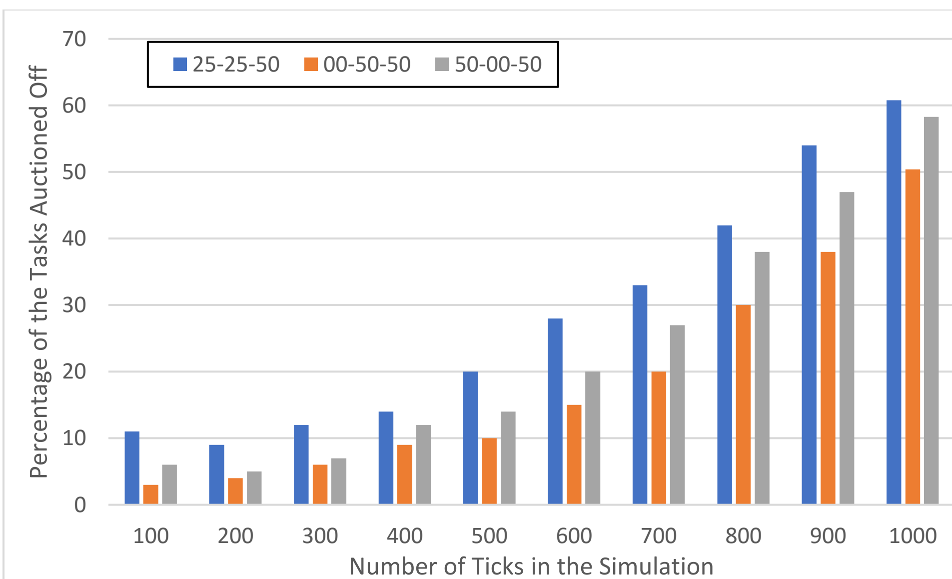


Figure 4.3. Average percentage of tasks auctioned off for each of the three AD configurations (25-25-50, 00-50-50, and 50-00-50) every 100th tick, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

Team Makeup over Time. Here we want to investigate how the role of each agent type changes over time in the teams formed. In the following, Figures 4.4-4.6 display team makeup for the average number of tasks auctioned off over time for the three AD configurations, respectively. Tables 4.9-4.11 show the actual numbers correspondingly.

Tick	G only (%)	S only (%)	A only (%)	Homogeneous (%)	G and S only (%)	G and A only (%)	S and A only (%)	G, S and A (%)	Heterogeneous (%)	Total
100	12288.20 (27.42%)	6275.85 (14.00%)	7349.76 (16.39%)	25913.81 (57.82%)	6301.39 (14.06%)	5890.51 (13.14%)	4564.99 (10.18%)	2144.21 (4.78%)	18901.1 (42.17%)	44814.91
200	7792.41 (20.17%)	7931.08 (20.53%)	6566.04 (16.99%)	22289.53 (57.70%)	7486.17 (19.38%)	2429.62 (6.28%)	2078.61 (5.37%)	4344.68 (11.24%)	16339.08 (42.29%)	38628.61
300	8013.34 (17.16%)	7492.21 (16.04%)	6702.66 (14.35%)	22208.21 (48.63%)	10403.68 (22.28%)	2523.00 (5.40%)	2627.88 (5.62%)	8921.82 (19.11%)	24476.38 (52.42%)	46684.59
400	7369.37 (13.63%)	7148.85 (13.22%)	5324.07 (9.85%)	19842.29 (36.71%)	12562.71 (23.24%)	3789.67 (0.70%)	2586.91 (4.78%)	15260.70 (28.23%)	34199.99 (63.28%)	54042.28
500	5793.18 (9.14%)	6753.64 (10.65%)	4553.00 (7.18%)	17099.82 (26.98%)	16404.76 (25.88%)	4809.01 (7.58%)	2652.78 (4.18%)	22401.63 (35.35%)	46268.18 (73.01%)	63368.00
600	6057.59 (7.96%)	6506.38 (7.95%)	5839.08 (7.67)	18403.05 (24.18%)	20425.97 (26.84%)	5248.33 (6.89%)	3113.48 (4.09%)	28901.16 (37.98%)	57688.94 (75.81%)	76091.94
700	5110.80 (0.60%)	5427.73 (6.38%)	5635.17 (6.62%)	16173.7 (19.01%)	23220.45 (27.30%)	6279.77 (7.38%)	3706.69 (4.35%)	35655.39 (41.93%)	68862.30 (80.98%)	85035.47
800	6171.23 (5.98%)	6206.60 (6.02%)	5151.34 (4.99%)	17529.17 (17.01%)	27765.04 (26.94%)	8746.06 (8.48%)	4651.42 (4.51%)	44350.71 (43.04%)	85513.23 (82.98%)	103042.4
900	7102.48 (6.04%)	6222.44 (5.29%)	6506.86 (5.54%)	19831.78 (16.88%)	31892.76 (27.16%)	9702.56 (8.26%)	4983.64 (4.42%)	51012.75 (43.44%)	97591.71 (83.11%)	117423.49
1000	6745.91 (5.23%)	6539.28 (5.07%)	6725.02 (5.21%)	20010.21 (15.51%)	34888.65 (27.05%)	9429.84 (7.31%)	5835.93 (4.52%)	58787.03 (45.58%)	108941.45 (84.48%)	128951.66

Table 4.9. Average types of agents forming teams together for AD 25-25-50 configuration, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

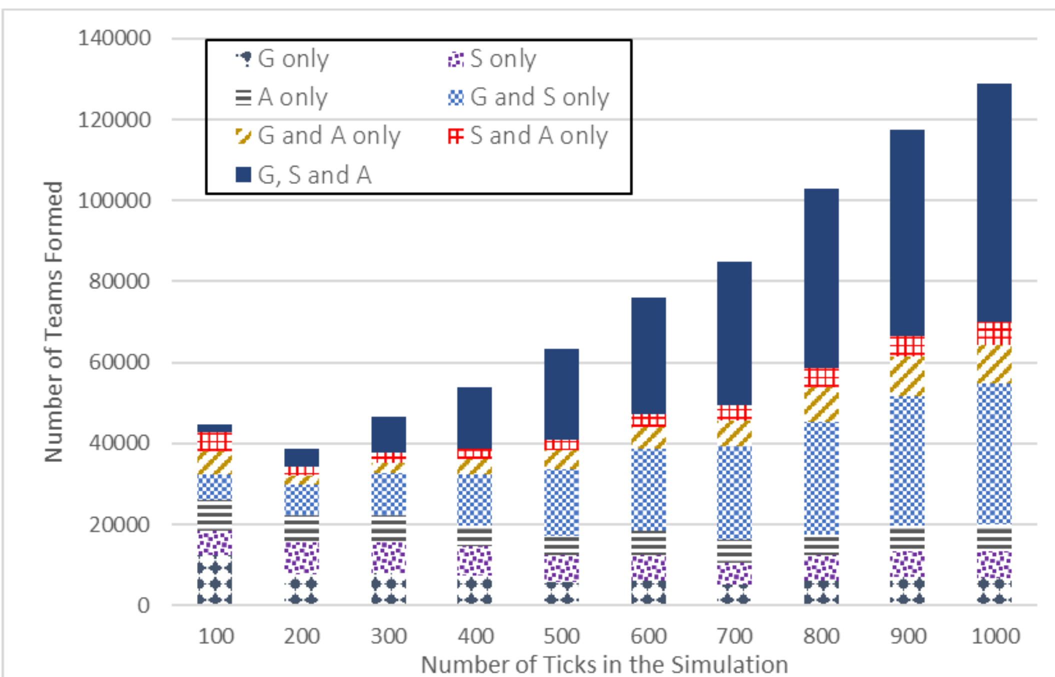


Figure 4.4. Average (over 100 runs) number of teams—with different combinations of agent types—formed for the AD 25-25-50 configurations, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

Tick	G only (%)	S only (%)	A only (%)	Homogeneous (%)	G and S only (%)	G and A only (%)	S and A only (%)	G, S and A (%)	Heterogeneous (%)	Total
100	0	6259.21 (34.94%)	6147.04 (34.31%)	12406.25 (69.53%)	0	0	5507.44 (30.47%)	0	5507.44 (30.47%)	17913.69
200	0	8530.65 (49.25%)	5485.32 (31.67%)	14015.97 (80.93%)	0	0	3303.42 (19.07%)	0	3303.42 (19.07%)	17319.39
300	0	7640.19 (45.35%)	4955.86 (29.41%)	12596.05 (74.78%)	0	0	4520.11 (25.22%)	0	4520.11 (25.22%)	17116.16
400	0	6729.11 (42.60%)	3787.43 (23.97%)	10516.54 (66.59%)	0	0	5276.86 (33.41%)	0	5277.86 (33.41%)	15794.4
500	0	6440.92 (41.51%)	3127.29 (20.15%)	9568.21 (61.68%)	0	0	5946.39 (38.32%)	0	5946.39 (38.32%)	15514.6
600	0	6270.01 (35.55%)	3919.16 (22.22%)	10189.17 (57.78%)	0	0	7446.60 (42.22%)	0	7447.60 (42.22%)	17636.77
700	0	6070.77 (28.80%)	3410.84 (16.18%)	9481.61 (44.99%)	0	0	11596.10 (55.01%)	0	11596.10 (55.01%)	21077.71
800	0	7567.10 (27.46%)	3494.22 (12.68%)	11061.32 (40.15%)	0	0	16493.42 (59.85%)	0	16493.42 (59.85%)	27554.74
900	0	8613.44 (24.42%)	5491.30 (15.57%)	14104.74 (40.00%)	0	0	21160.13 (60.00%)	0	21160.13 (60.00%)	35264.87
1000	0	9855.87 (24.12%)	5001.70 (12.24%)	14857.57 (36.38%)	0	0	25993.25 (63.62%)	0	25994.25 (63.62%)	40851.82

Table 4.10. Average types of agents forming average number of teams together for AD 00-50-50 configurations, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

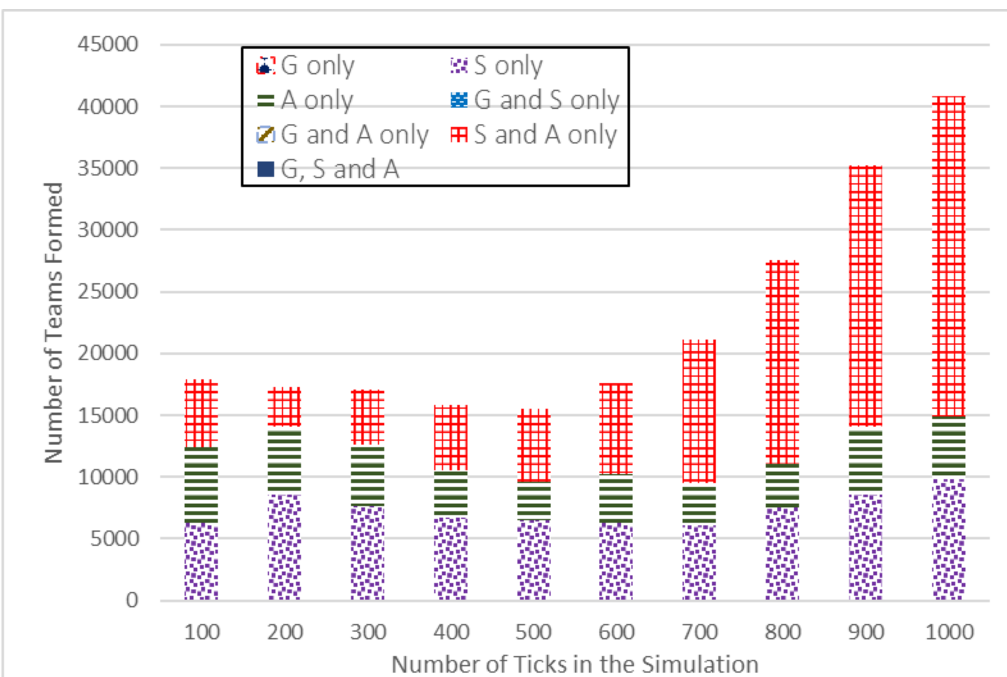


Figure 4.5. Average types of agents forming average number of teams together for AD 00-50-50 configurations, with TD Medium, and AO = TO = 0, $p < 0.0001$.

Tick	G only (%)	S only (%)	A only (%)	Homogeneous (%)	G and S only (%)	G and A only (%)	S and A only (%)	G, S and A (%)	Heterogeneous (%)	Total
100	15068.23 (52.76%)	0	6013.00 (21.05%)	21081.23 (73.83%)	0	7476.53 (26.17%)	0	0	7476.53 (26.17%)	28557.76
200	10336.00 (52.48%)	0	5384.12 (27.33%)	15720.12 (79.82%)	0	3975.21 (20.18%)	0	0	3975.21 (20.18%)	19695.33
300	10719.65 (49.44%)	0	5327.87 (24.57%)	16047.52 (74.03%)	0	5631.19 (25.97%)	0	0	5631.19 (25.97%)	21677.71
400	10531.80 (44.87%)	0	4165.34 (17.74%)	14697.14 (62.63%)	0	8772.94 (37.37%)	0	0	8772.94 (37.37%)	23470.08
500	11159.11 (40.93%)	0	3393.87 (12.44%)	14552.98 (53.38%)	0	12710.14 (46.62%)	0	0	12710.14 (46.62%)	27263.12
600	13578.39 (39.15%)	0	4597.00 (13.25%)	18175.39 (52.42%)	0	16502.29 (47.58%)	0	0	16502.29 (47.58%)	34677.68
700	13767.66 (33.72%)	0	3888.17 (9.52%)	17655.83 (43.26%)	0	23164.32 (56.74%)	0	0	23164.32 (56.74%)	40820.15
800	18364.93 (35.45%)	0	3902.92 (7.53%)	22267.85 (42.99%)	0	29534.21 (57.01%)	0	0	29534.21 (57.01%)	51802.06
900	21179.22 (33.09%)	0	5529.32 (8.64%)	26708.54 (41.74%)	0	37283.93 (58.26%)	0	0	37283.93 (58.26%)	63992.47
1000	23019.98 (32.25%)	0	5294.39 (7.41%)	28314.37 (39.68%)	0	43043.35 (60.32%)	0	0	43043.35 (60.32%)	71357.72

Table 4.11. Average Types of agents forming teams together for AD 50-00-50 configurations, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

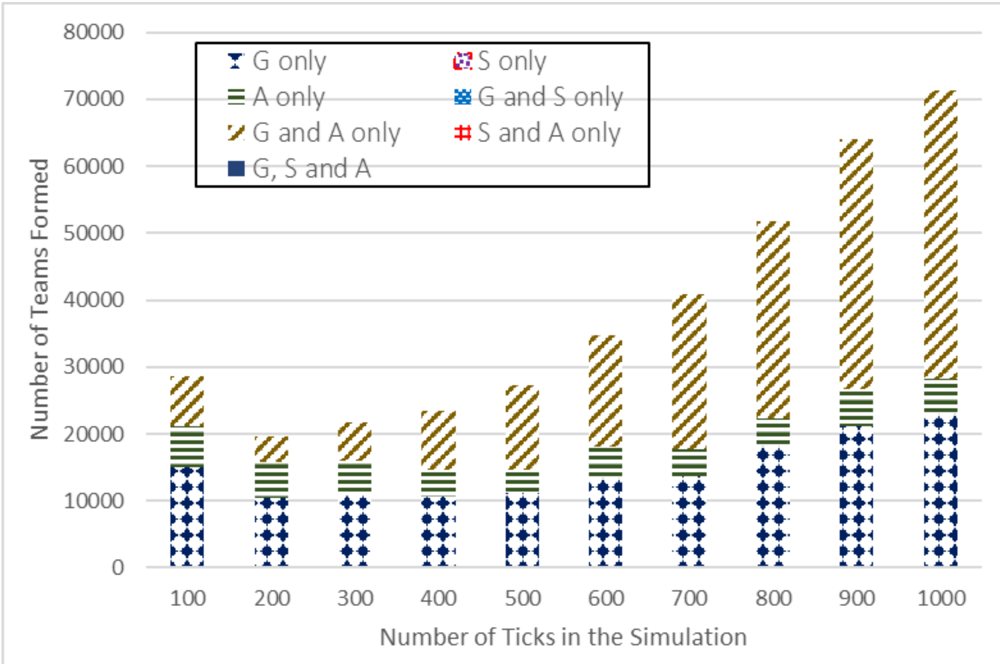
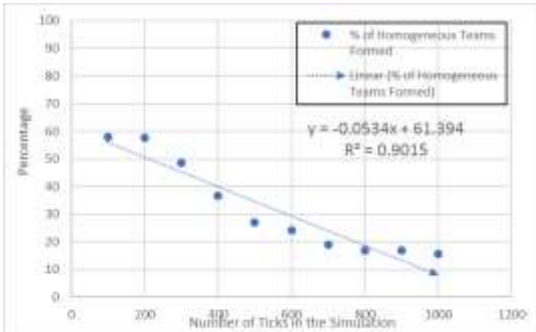


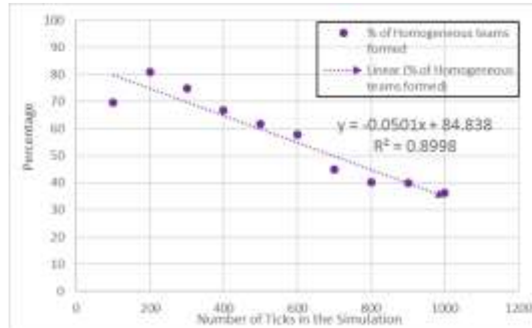
Figure 4.6. Average types of agents forming average number of teams together for AD 50-00-50 configurations, with TD = Medium, and AO = TO = 0, $p < 0.0001$.

As seen from Figures 4.4, 4.5 and 4.6 as the simulation progressed, the agents started forming more heterogeneous teams than homogeneous teams. For example, in AD 25-25-50, the average percentages of homogeneous and heterogeneous teams were 57.82% and 42.17% at the 100th tick of the simulation and over time, they changed to 26.98% and 73.01% at the 500th tick, and eventually reached 15.51% and 84.48% at the 1000th tick, respectively. We see similar trends for AD 50-00-50 and AD 00-50-50.

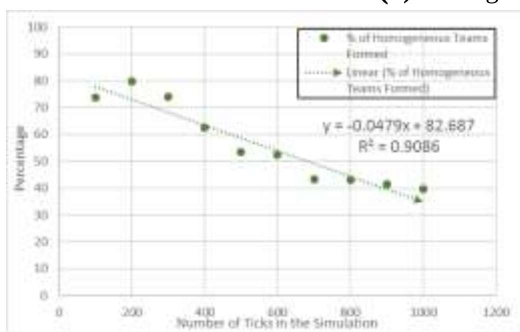
Figure 4.7 presents the linear regressions for each of the three AD configurations for the average percentage of the number of homogeneous and heterogeneous teams formed timely.



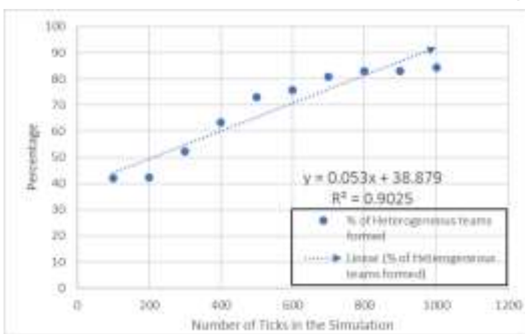
(a) Homogeneous AD 25-25-50



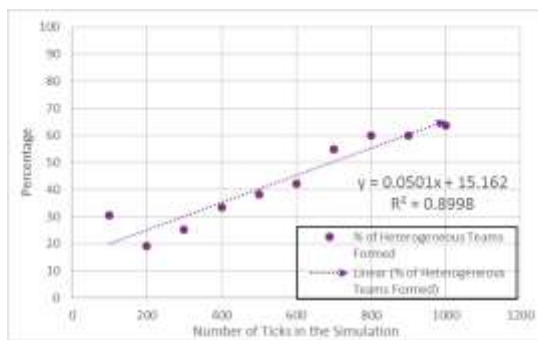
(b) Homogeneous AD 00-50-50



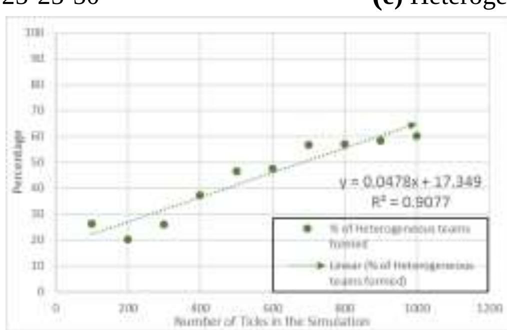
(c) Homogeneous AD 50-00-50



(d) Heterogeneous AD 25-25-50



(e) Heterogeneous 00-50-50



(f) Heterogeneous 50-00-50

Figure 4.7. Linear Regression Plots for Average Percentages of Homogeneous and Heterogeneous teams formed for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

Diversity helps an agent bid for a wider variety of tasks as well as learn from a wider skillset and be able to contribute more beneficially to the environment as compared to a homogeneous agent group. The increase in the heterogeneous teams formed shows that it is more beneficial to have a diverse agent skillset as compared to a homogeneous group of agents, to evolve the agents through learning by observation. Also, in a diverse group an agent learns through observation and doing, there is a rise in the level of expertise in the skillset, which in return equips the agents to be able to bid for a variety of subtasks and increase the chances of the subtask being allocated to an agent. To further investigate this, Figure 4.2 shows the average learning by observation gain for the average percentage of homogeneous versus the heterogeneous teams formed for all the three AD configurations. Indeed, as seen from Figure 4.2, the average learning gain for heterogeneous teams was higher than that for homogeneous teams, for all three AD configurations. Thus, it is also possible for agents to bid for tasks that require more diverse capabilities—involving heterogeneous members—in order to reap benefits from learning, and thereby forming more heterogeneous teams over time.

The above understanding leads us to expect that the majority of the population of the team should be composed of Generalists. As there is a wider variety of capabilities at a decent level of expertise to learn from if there are majority Generalists, and this majority agent type could be influencing the learning by observation of the other teammates as well. This helps the entire team evolve as a whole, since Apprentices can learn a wider variety of skills from Generalists, Generalists can learn from other Generalists as well, and Specialists can learn from Generalists that have qualified level of expertise that Specialists might not have.

To further investigate into this, Figure 4.8 presents the dominant agent type present in each team for all the three AD configurations. As seen from Figure 4.8, in a diverse environment composed of all three types (G, S, A), more number of teams are dominated by Generalists. As Generalists are the dominant agent types over time, Generalists get to learn from other Generalists while Apprentices also get to learn a variety of skillsets from the Generalists and get better. This backs up our expectation of Generalists being the dominant agent types and contributing to the timely increase of heterogeneous teams and then the decrease of homogeneous teams.

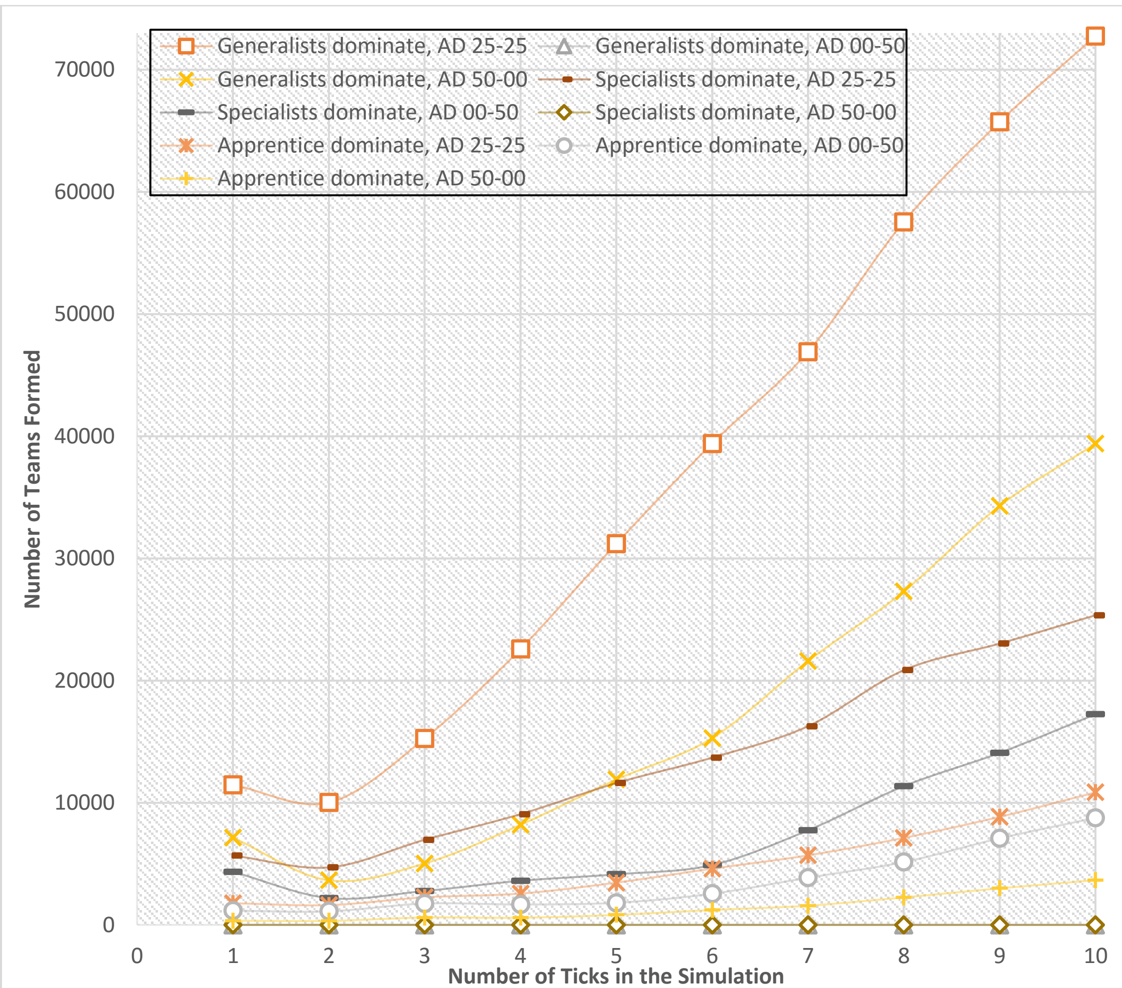


Figure 4.8. Average total number of teams under dominant Agent type for each of the three AD configurations (25-25-50, 00-50-50, 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

4.3.2 Impacts of Learning and Agent Diversity

Here we investigate how an agent changes its type as a result of learning by doing and learning from observation, and if agent diversity has an impact on this learning. For our simulation, recall that an agent qualifies as a Specialist if at least 1/5 of their skills are above 0.85 level of expertise, for a Generalist 3/5 of their capabilities are above 0.5 level of expertise. Apprentices have a 0.15 mean of distribution for all the levels of expertise of their capabilities, as described in detail in Section 4.1 and Chapter Methodology. When an agent changes its type, we define this as "promotion".

For example, say a Generalist agent has the following capabilities and levels of expertise: 0.5, 0.6, 0.7, 0.3, 0.2, through learning, the agent's updated capability vector now is 0.5, 0.6, 0.9, 0.4, 0.2. According to our definition of a Generalist and a Specialist, the agent now is both a Specialist and a Generalist. We deem this phenomenon as a type promotion. Likewise, an Apprentice agent can become either a Specialist or a Generalist or both given enough experience of working on teams to solve tasks. However, it is not possible for a Specialist to become a non-Specialist.

4.3.2.1 Promotion Statistics and Diversity

This section presents the promotions for all three AD configurations (25-250-50, 00-50-50, and 50-00-50) and TD = Medium, and TO = AO = 0 and helps us understand if AD has an impact on the promotion of agents.

Overall Statistics. Table 4.12 presents the average percentage of the promoted and non-promoted agent types for each of the three AD configurations (the values are normalized by the count of each agent type, since there are configurations when one agent type may have more agents than the other).

	25-25-50	00-50-50	50-00-50
% A→ only G	0.0025	0.0018	0.0027
% A→ only S	00.8570	00.1708	00.3082
% A→ both {S,G}	99.1292	99.8200	99.6749
% G→S	97.7638	0	97.9222
% Non-promoted G	1.07322	0	1.0307
% Non-promoted S	100	100	0
% Non-promoted A	0.0112	0.0072	0.0096

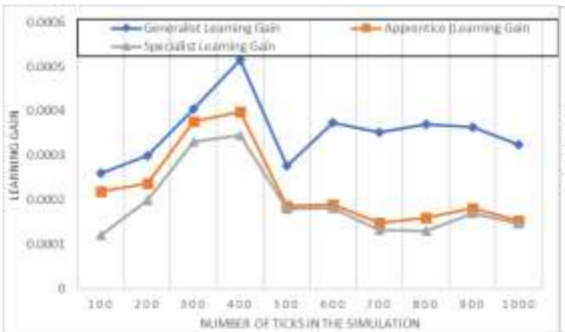
Table 4.12. *Average (over 100 runs) percentages of promoted and non-promoted agents* for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, standard error = 0.0003.

As seen from Table 4.12 it can be seen that Apprentices get the most number of promotions. This is because this is the agent type at a lower level of expertise as compared to the other two agent types and they can learn the most by observing the other two types. Generalists already have a decent level of expertise as compared to an Apprentice, Generalists can learn by observing their fellow Generalists or Specialists and improve by learning through observation.

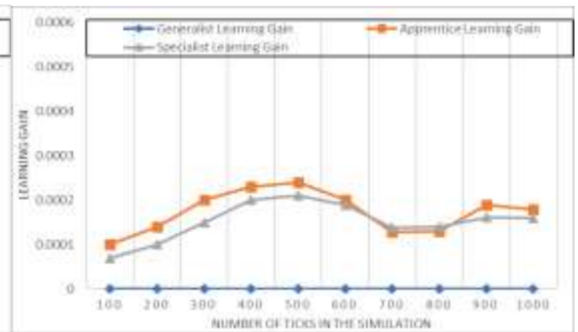
We also see that the number of Apprentices that were not promoted is smaller for the 00-50-50 configuration than it is for the 50-00-50 configuration (though the difference is not high). This hints that Apprentices tend to improve more in the presence of Generalists than Specialists because it is easier for Apprentices to learn a wider variety of capabilities from a Generalist than a Specialist, and thus Generalists are important for bridging the gap between Apprentices and Specialists.

4.3.2.2 Learning Gains over Time.

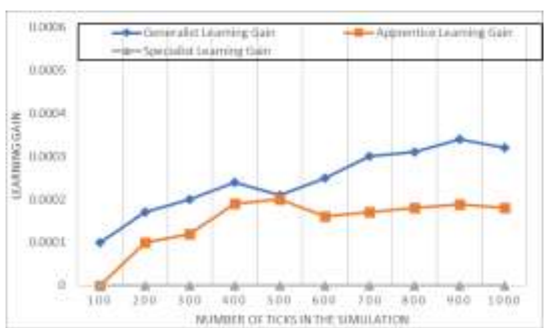
This helps us analyze if agent diversity has an impact on the learning of the agents in a team and in return help the agent types evolve through learning by doing and observation from each other. Figure 4.9 presents the average learning gain plots for all three agent types for all three AD configurations with TD = Medium, and no openness.



(a) AD 25-25-50



(b) AD 00-50-50



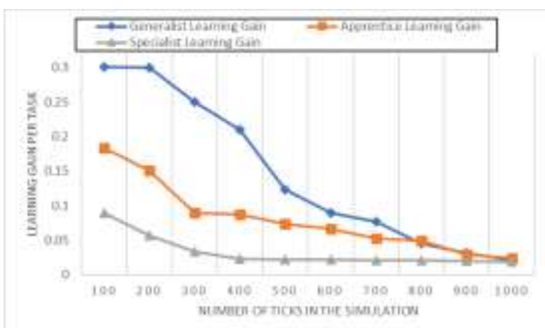
(c) AD 50-00-50

Figure 4.9. Average (over 100 runs) learning gains per tick for all three agent types, for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

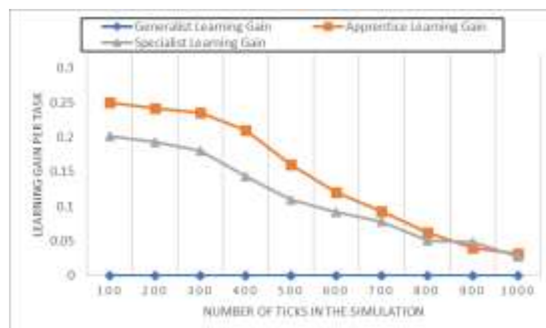
First, as seen from Figure 4.9 *the average learning gain is the highest for AD 25-25-50, followed by 50-00-50 then 00-50-50*. This indicates that a more diverse environment helps produce higher learning gains. We see better gains in a 25-25-50 environment than the other environments because the Apprentices were able to learn a wider variety of tasks from the Generalists. Once they had learned a level of expertise from Generalists beyond which they no longer learn anything new, they could then start learning from Specialists and continue to get better. Thus, the Generalists helped bridge the gap between the Apprentices and the Specialists. In the other configurations, absence of any agent type hampers this balance. In the 00-50-50, the Apprentices, in the absence of the Generalists, take longer to reach a level where they can start learning from Specialists. Once they reach a point where Apprentices start learning the few capabilities the Specialists are good at. However, since there are only a few capabilities that the Specialists are good at, the Apprentices and Specialists reach a point where they do not learn anything new. This also explains why the curves for 25-25-50 and 00-50-50 tend to stabilize after going up and then coming down, but for the 50-00-50 (only Apprentices and Generalists), the curve does not stabilize after going up and coming down. The Apprentices and Generalists in the latter kept learning because they did not reach a point at which the agents had gotten good at capabilities such that their learning gains began to diminish and converge with that of the agents it learned from. Generalists kept observing the other Generalists and getting good at a wider variety of capabilities; similarly, Apprentices also kept observing the Generalists as well as the other Apprentices and kept sharpening their capabilities. Also, the Generalists could observe and learn from the Apprentices in the environment for the few capabilities that the Generalists were not good at. Thus, the agents' learning gains kept rising as they had not reached at a same level of expertise as their counterparts. The curve for the 25-25-50 environment stabilized the smoothest and then for 00-50-50. A reason for this is that the agents gained the same level of knowledge in the 25-25-50 environment causing the learning gains of the agents to converge and hence the curve stabilizes. Also, the learning gains converge at an earlier tick for the 25-25-50 as compared to the 00-50-50, this also backs the fact that diversity helps the agents learn faster and evolve by getting good at their capabilities.

Second, it can be seen that the learning gain for the Generalists is the most in the environment. But the learning gain is lesser for them in the 50-00-50 as compared to the 25-25-50 environment because they do not have the Specialists to help them evolve more effectively further.

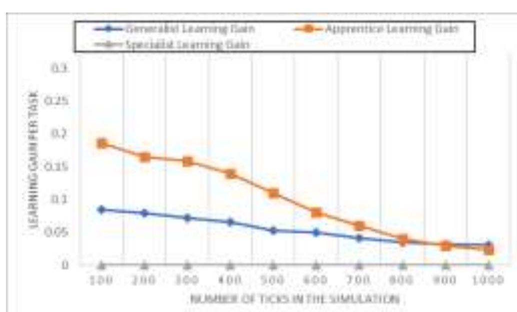
Third, we take a look at the average learning gains per task to evaluate the efficiency with which the agents learned in each AD configuration. Figure 4.10 shows the average learning gains per task for each of the three AD configurations.



(a) AD 25-25-50



(b) AD 00-50-50



(c) AD 50-00-50

Figure 4.10. Average (over 100 runs) learning gains per task for all three agent types, for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

As can be seen in the 25-25-50 configuration, the Generalists gained the highest learning because initially they were eligible for bidding and winning a wider variety of tasks, as the Specialists were good at only very few capabilities and Apprentices hardly had any decent level of expertise in any of their capabilities. Thus, the Generalists showed the highest learning gain curve, as they had a chance to sharpen a wider variety of capabilities by performing a variety of tasks. The Specialists, on the other hand, tended to bid for the same type of tasks utilizing the same set of their capabilities, since they were good at only select few capabilities, which in turn reduced their learning gain from each task. In the 00-50-50 configuration, the Apprentices showed the highest learning gain as they had larger numbers of capabilities at a low level of expertise which they kept improving. Also in the absence of Generalists, the Specialists were able to bid on a variety of tasks and kept increasing their learning gain. In the 50-00-50 configuration, on account of the absence of the Specialists as teammates, the Generalists did not gain much in their learning from each task. However, the Apprentices on the other hand kept learning from Generalists and showed a larger learning gain consistently over time until the Apprentices gained enough expertise to render learning non-effective towards the end of the simulation. Another observation is that the learning gain of the Specialists was higher in the 00-50-50 environment as compared to the learning gain of the Generalists in the 50-00-50 environment. Because the Specialists were good at very few capabilities as compared to the Generalists who were good at a wider variety of capabilities, the Specialists had more diverse "rooms" to improve as compared to the Generalists, even just learning from the other Apprentices the capabilities in which the Specialists had very low level of expertise. This makes the learning gain of the Specialists higher than that of the Generalists.

4.3.3 Discussions

We have investigated the impacts of agent diversity on team formation and team makeup (Section 4.3.1), and on the impacts of agent diversity on learning (Section 4.3.2). In particular, we investigated teammate statistics and how they evolved over the duration of the simulation under different levels of agent diversity in Section 4.3.1, and how agents' levels of expertise in their capabilities changed over time—leading to promotion from one agent type to another—and the effectiveness and efficiency of their learning in terms of learning gains over time, in Section 4.3.2.

Here we summarize the key findings and implications:

(1) From Sections 4.3.1.1 and 4.3.1.2, we reported that the average number of teams formed (and thus the average number of tasks auctioned off) was highest for the AD = 25-25-50 configuration, considered to be the most diverse among the three configurations used in our experiment. The better performance by the 25-25-50 configuration was attributed to the presence of all three agent types (i.e., Specialists, Generalists, and Apprentices) causing the following to happen:

(a) A combination of Specialists—for their "high-level" expertise—and Generalists—for their "wide" range of decent-level capabilities—allow more tasks to be auctioned off, providing more opportunities for agents to learn;

(b) The three agent types learned more effectively (e.g., average learning gain per tick highest for AD = 25-25-50) and more efficiently (e.g., average learning gain per task highest for AD = 25-25-50) because of the reduced learning gap as modeled by Equation (3.3); this is because all three agent types have a key role to play in the environment, and without any agent type a balance cannot be established. Apprentices are a major learning force in the environment, the "next-generation" of the environment. The Generalists are the agent type which helps bridge the gap between the Apprentices and the Specialists. They are the agents that bring a wider perspective to the environment. The Specialists bring the highest level of expertise (in few capabilities) to the environment than do the other two types. The Generalists need the Specialists to sharpen their expertise else their learning will come to a stagnant still; and

(c) As agents gained in expertise in their capabilities, they were able to form more teams to complete tasks. From this Investigation, there is clear evidence that Agent Diversity has an impact on the team-formation and tasks auctioned off as diversity helps the agents learn and flourish and accomplish more tasks.

(2) From Section 4.3.2.1, we observed the impact of learning on agent types as we computed the "type promotion" occurrences. Again, the 25-25-50 configuration (the most diverse one) yielded the highest average number of type promotions. First, the Apprentices were found to be the major learning force on account of their low level of expertise for all of their capabilities. Second, the number of promoted Apprentices was the highest in the 25-25-50 configuration. Again, this is due to the human learning model that produces the largest learning gain when the teacher and learner agents' levels of

expertise are not too close or too far apart, and that the 25-25-50 configuration facilitated that. This thus confirms the above findings.

(3) From Section 4.3.2.2, we observed the average learning curves of the agents in the three AD configurations. We reported that all configurations yielded average learning curves of similar slopes but the 25-25-50 configuration had a higher starting point due to the advantage of being able to complete more tasks to begin with. Again, we confirm that agent diversity plays an important role in learning as well as for improving agent promotions.

4.4 Investigation 2: Impacts of Task Diversity

In this section, we investigate whether and how Task Diversity (TD) has an impact on the team formation by analyzing the average percentage of tasks auctioned off (i.e., completed).

4.4.1 Tasks Auctioned Off over Time and Task Diversity

To be able to focus on task diversity, we set openness to zero and consider only one AD configuration. Figure 4.11 shows the average percentage of tasks auctioned off for TD = Low, Medium, and High, with AD = 25-25-50, AO = TO = 0.

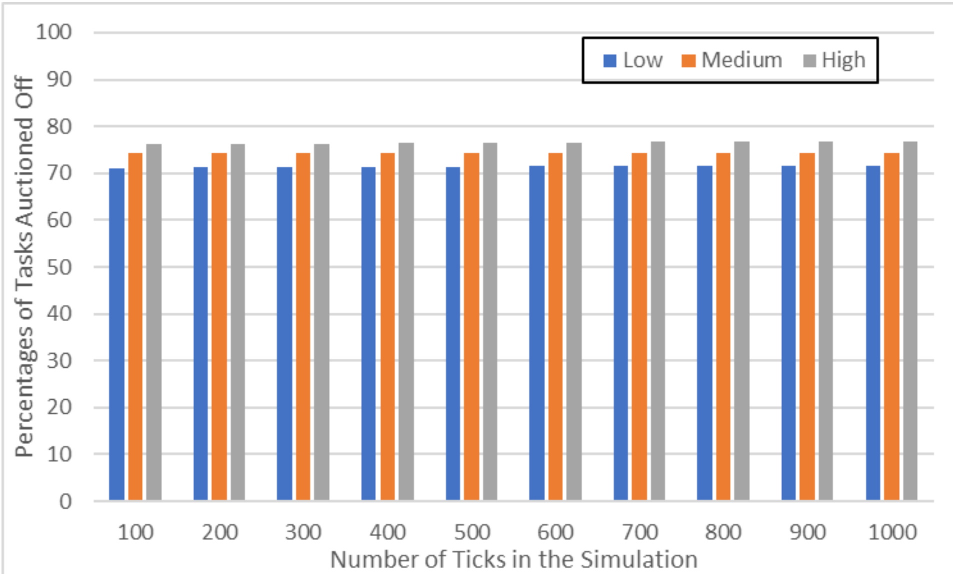
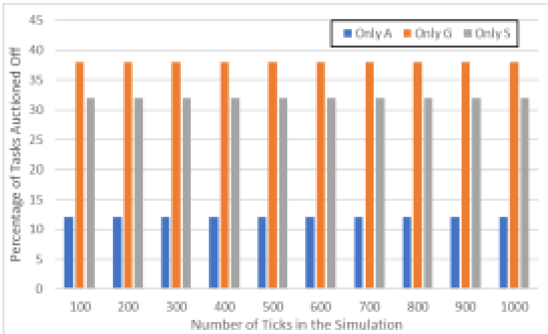


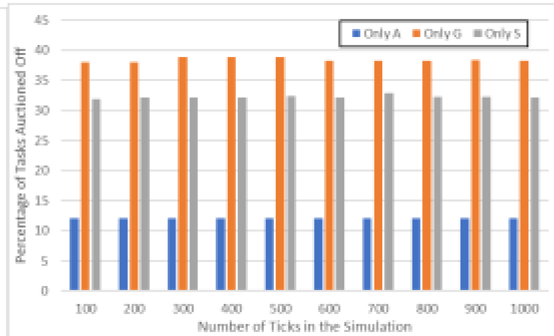
Figure 4.11. Average (over the 100 runs) percentages of tasks auctioned off for TD = Low, Medium, and High, with AO = TO = 0, and AD = 25-25-50, $p < 0.0001$.

It can be seen from Figure 4.11 that the number of tasks auctioned off was the lowest for low TD (just above 70%). One reason for this is that since AO and TO were both zero, agents who always ended up winning their bids kept winning throughout all auction rounds. The average percentage of tasks auctioned off increased with the increase in the task diversity level.

Figure 4.12 shows the average percentage of tasks auctioned off for TD = Low, Medium, High, with AO = TO = 0 and, only Apprentices (Only A), only Generalists (Only G), and Only Specialists (Only S) present. This investigation helps us further analyze each agent type separately.

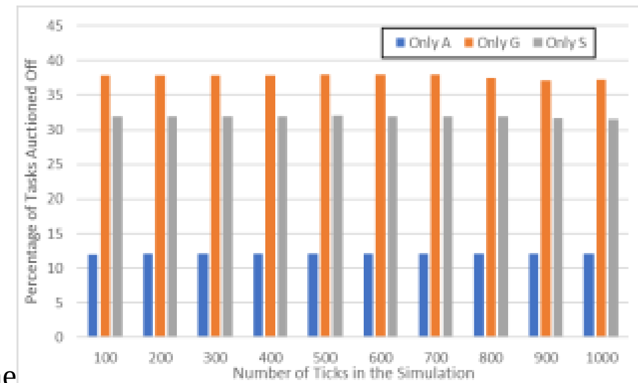


(a) TD = Low



(b) TD = Medium

Gene



(c) TD = High

Figure 4.12. Average (over 100 runs) percentage of tasks auctioned off for TD = Low, Medium, and High for AD = Only A, Only G, and Only S, with AO = TO = 0, $p < 0.0001$.

As can be seen from Figure 4.1, the Generalists achieved a higher average percentage of tasks auctioned off than did the other two agent types. This was because the Generalists had a wider variety of capability set than the other two types, making the Generalists eligible to bid for a wider variety of tasks. The Apprentices secured the least number of tasks since these agents had low level of expertise in all their capabilities. Thus, it became more difficult for them to be able to bid for tasks.

One possible reason for the low average percentages of tasks getting auctioned off could be on account of insufficient number of agents casting their bids for the task. A probable reasoning for this insufficiency could be because agents may have been bidding for different tasks, and even if they were winning bids, they did not have sufficient number of teammates. However, all these agents that won bids but still never got tasks assigned could have worked together and completed more tasks, instead of missing an opportunity by bidding on different tasks. Thus, we calculate the average number of missed opportunities by matching the capabilities of the agents that won bids but did not get tasks assigned with the capability threshold requirements of tasks not auctioned off. An opportunity is considered to be a ***missed*** opportunity if an agent won a bid and the task was not auctioned off on account of insufficient teammates. However, these unfilled teammate posts could have been filled up by the other agents in the environment who won their bids as well but their tasks were not auctioned off for the same reason. Table 4.13 shows the average percentages of missed opportunities for all three TD configurations (Low, Medium, and high) for AD = Only A, Only G and, Only S, with TO = AO = 0.

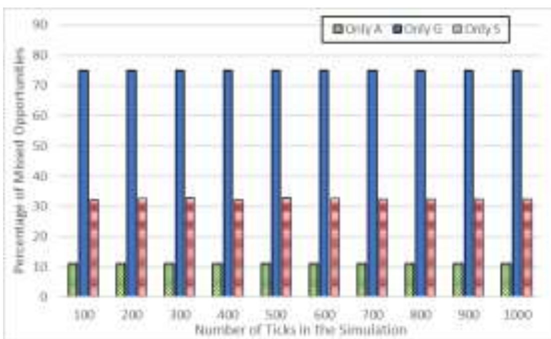
	Low TD	**Medium TD**	**High TD**
Only A	11.1354%	12.0210%	13.9830%
Only G	78.8921%	79.5736%	79.9892%
Only S	34.3947%	35.2857%	38.2191%

Table 4.13. Average (over 100 runs) percentages of missed opportunities for each of the three TD configurations (Low, Medium, and High), for Only A, Only G, and Only S, with AO = TO = 0, standard error = 0.0004.

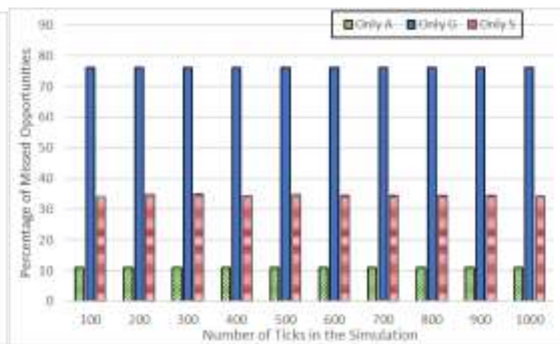
It can be seen that as the level of Task Diversity increases, the average percentage of missed opportunities increases as well. Also, the average percentage of missed opportunities is the highest for the Generalists, followed by the Specialists and then the Apprentices. Of course, as the Generalists have more diverse capabilities in which they have decently fit level of expertise, they also become eligible to bid for a wider variety of tasks. Thus, instead of them going for the same tasks, they are more likely to split across by bidding on different tasks.

Therefore, even though there were sufficient capabilities in the agent population, and there were agents that ended up winning bids, some tasks never got auctioned off because there were not sufficient number of Generalists bidding for a given task. Having more Generalists tended to hamper the tasks that could have been auctioned off. Had the Generalists channeled their expertise together on the same tasks, more tasks would have auctioned off instead of splitting themselves over different tasks and resulting in lesser tasks running to completion. The same logic applies to the other two agent types as well. However, since they were not at a decent level of expertise for such a wide variety of capabilities, their missed opportunity average percentages were lower than that of Generalists.

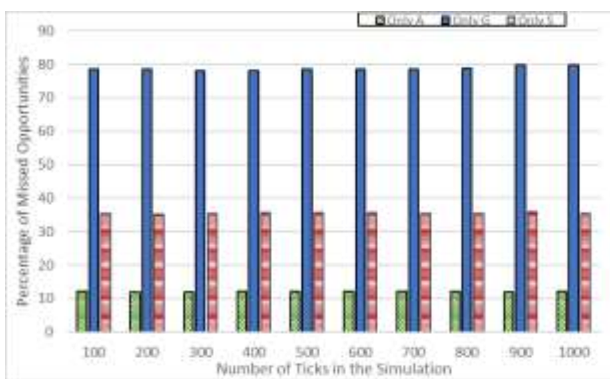
This brings us to another point. Our equations and modeling of human learning in Section 3.1.2 were designed such that agents learn what to bid for and what not to bid for over time. That is, if a Generalist continues to win its bids for a task *T* but realizes that *T* is never getting auctioned off, will it learn to not bid for *T*? Figure 4. shows over time the average percentages of missed opportunities every 100th tick of the simulation, for all three TD configurations (Low, Medium, and High) for Only A, Only G, and Only S, with AO = TO = 0. It can be seen that the percentages of missed opportunities remained almost the same throughout the 1000 ticks for each of the three TD configurations. One reason for this could be that there were many permutations possible for agents to pursue another task. Because of this reason, agents would still end up with insufficient number of teammates. To make sure that the agents were attempting to bid for different tasks in case of task not being auctioned, we carried out analysis to calculate the average percentage of agents that submitted bids for different tasks in case they won a bid but the task was never auctioned off. We realized that for all three TD configurations (Low, Medium, and High) for each of the agent type, ~90% of the agents did bid for different tasks during the next auction round.   But this did not lead to higher success in terms of the number of tasks assigned, and did not stop the vicious cycles of bidding for another task during the next auction round.



(a) TD = Low



(b) TD = Medium



(c) TD = High

Figure 4.13. Average (over 100 runs) percentage of Missed opportunities for all three TD configurations (Low, Medium, and High), for AD = Only A, Only G, and Only S, with TO = AO = 0, $p < 0.0001$.

4.4.2. Discussions

To analyze the impacts of task diversity on the tasks that were auctioned off as well as agent learning, we looked at the average percentage of tasks auctioned off and the average learning gains for the agents for all three levels of Task Diversity (Low, Medium, and High).

The key observations and findings were:

(1) We saw that higher average percentage of tasks was auctioned off with an increase in the level of Task Diversity. This occurred because low Task Diversity causes the same teammates to keep coming together again and again to perform similar tasks and this restricts the capabilities they get to sharpen and the experience they gain by working on more varied tasks. On the other hand, higher levels of TD have more diversified tasks which presents the agents with the opportunity to work on a wider variety of tasks and hence learn more capabilities and in return causing more tasks to be auctioned off. Section 4.4.1 backed the investigation of Task Diversity having an impact on the team formation by providing evidence for the higher average percentage of tasks auctioned off with an increase in the level of Task Diversity.

(2) We learnt that the average percentage of missed opportunities increase with the increase in the level of Task Diversity. The missed opportunities were because there weren't insufficient number of teammates for the tasks to be auctioned off. The insufficiency arose because, instead of channeling their bids on the same tasks, the agents are split across different tasks as they bid for different tasks and even though they win

their respective bids, the tasks never get auctioned. Task Diversity has an impact on the missed opportunities and hence as seen earlier on the tasks auctioned off as well.

4.5. Comparing the Roles of Agent and Task Diversity

As seen earlier in Sections 4.3 and 4.4, both Agent and Task Diversity have an impact on the team formation and how agents learn to form these teams. Here we compare their impacts on the average percentage of tasks auctioned off and agents average learning gains, to analyze if either of Agent or Task Diversity has a greater impact on the team formation than the other.

4.5.1 Impacts on percentages of tasks auctioned off

From Figure 4.1 we plot the slopes for the average percentage of tasks auctioned off under each of the three AD (25-25-50, 00-50-50, and 50-00-50) and three TD (Low, Medium, and High) configurations with no openness. There is no openness since we wish to focus on the roles of diversity only. The *p*-values for all the AD and TD configurations (all 27 permutations) are less than 0.0001, which shows that the slopes of the all the three AD configurations are *statistically significantly different* as compared to the TD configurations. This shows that both AD and TD have different levels of impact on the percentage of tasks auctioned off. The slopes for all three AD configurations were higher than all those for the three TD configurations. This shows that the rate of tasks auctioned off is higher for Agent Diversity as compared to Task Diversity which shows that AD is more impactful than TD.

Agent Diversity has a greater impact on the team formation than Task Diversity. This is because even if there is no diversity amongst tasks in the environment, a diverse group of agents can learn to perform these tasks even if they have the basic expertise for it. Agents can learn by observation or learn by doing and keep sharpening their skills to get better at solving tasks. However, if there are tasks that require a high level of expertise and that level of expertise (for example, Specialists) is absent from the environment, then these tasks will not be auctioned off throughout the auction rounds. This is because neither the Apprentices nor the Generalists will ever be eligible for that task since they do not have that high level of expertise. Also, they cannot learn this high level of expertise since agents with that high level of expertise are absent. The same case follows for Generalists: in their absence, many tasks will not be auctioned off. This happens because Apprentices do not have anyone from which to learn a wide variety of tasks. Also on account of the learning gap between the Apprentices and the Specialists, it will take time for the Apprentices to start learning the few skills the Specialists are good at (and not a wide variety). Thus, in the presence of a diverse agent pool, it is easier to sharpen the level of expertise already present in the environment.

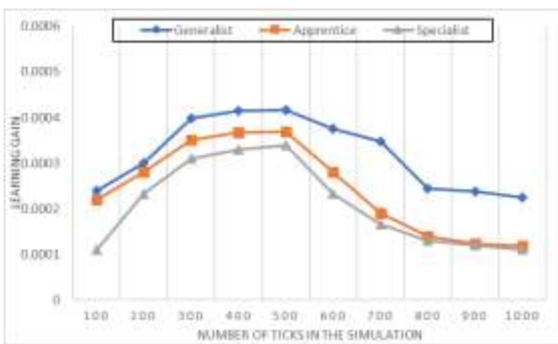
Agent Diversity (R-squared)			Task Diversity (R-squared)		
25-25-50	00-50-50	50-00-50	Low	Medium	High
5.9270 (0.9243)	4.9800 (0.8863)	5.8825 (0.9090)	0.0114 (0.3009)	0.0072 (0.4021)	0.0652 (0.9078)

Table 4.14. Slopes for average (over 100 runs) percentage of tasks auctioned off over time for all three AD (25-25-50, 00-50-5-, and 50-00-50) and TD (Low, Medium, and High) configurations, with AO = TO = 0 respectively. $p < 0.0001$

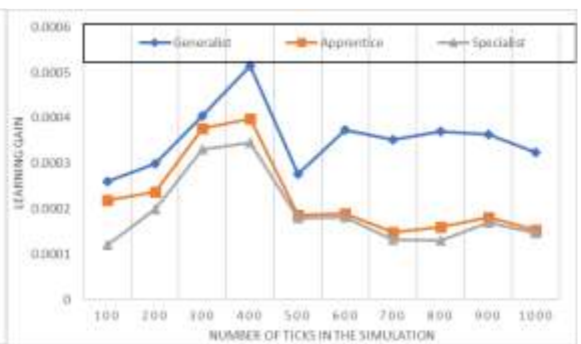
4.5.2 Impacts on learning gains

To compare the results of Agent Diversity and Task Diversity on the learning gains we picked the 25-25-50 AD configuration since this is our most agent-diverse environment, and applied TD = Low, Medium and High on this configuration to compare the learning gains.

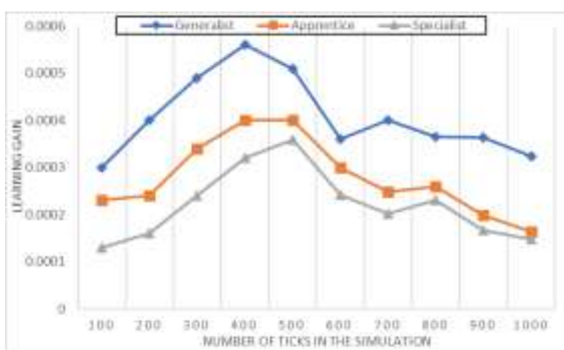
Figure 4.14(a)-(c) show the learning gains for AD = 25-25-50, TD = High, Medium, and Low, with AO = TO = 0, respectively. Again, there is no openness since we wish to focus on the role of diversity.



(a) TD = Low



(b) TD = Medium

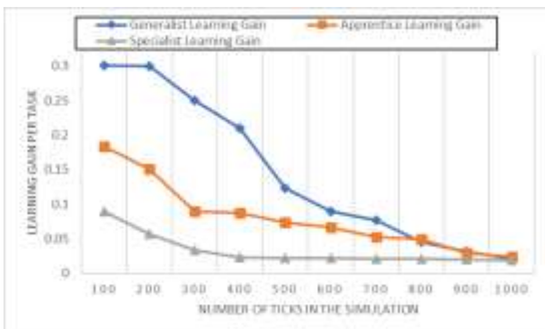


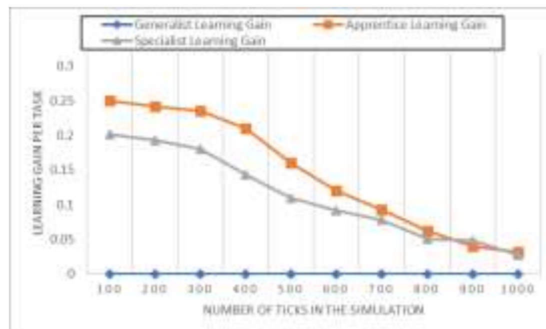
(c) TD = High

Figure 4.14. Average (over 100 runs) learning gains per tick for TD = Low, Medium, and High, AD = 25-25-50, with AO = TO = 0, $p < 0.0001$.

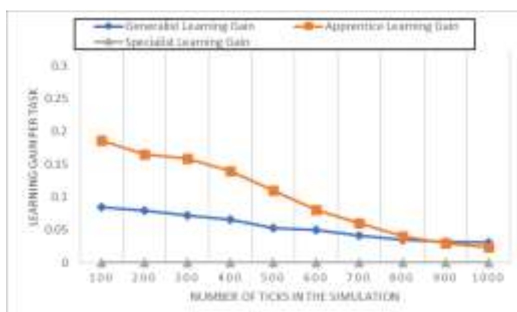
As Task Diversity increases, the average learning gains also increase. This is because in a more task-diverse environment, agents have an opportunity to work on a variety of tasks. Because of this opportunity, the agents keep learning a wide variety of capabilities and keep improving their learning gains.

Third, we take a look at the average learning gains per task to evaluate the efficiency with which the agents learned in each AD configuration. Figure 4.15 shows the average learning gains per task for each of the three AD configurations.





(a) AD 25-25-50 **(b)** AD 00-50-50



(c) AD 50-00-50

Figure 4.15. Average (over 100 runs) learning gains per task for all three agent types, for all three AD configurations (25-25-50, 00-50-50, and 50-00-50), with TD = Medium, and AO = TO = 0, $p < 0.0001$.

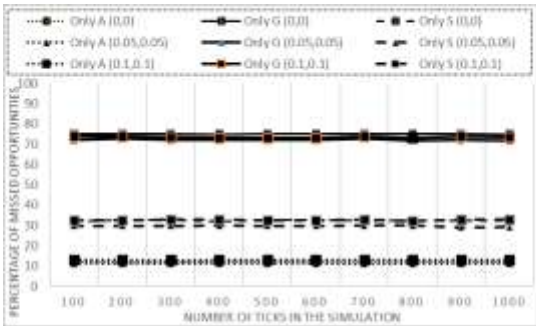
As can be seen in the 25-25-50 configuration, the Generalists gained the highest learning because initially they were eligible for bidding and winning a wider variety of tasks, as the Specialists were good at only very few capabilities and Apprentices hardly had any decent level of expertise in any of their capabilities. Thus, the Generalists showed the highest learning gain curve, as they had a chance to sharpen a wider variety of capabilities by performing a variety of tasks. The Specialists, on the other hand, tended to bid for the same type of tasks utilizing the same set of their capabilities, since they were good at only select few capabilities, which in turn reduced their learning gain from each task. In the 00-50-50 configuration, the Apprentices showed the highest learning gain as they had larger numbers of capabilities at a low level of expertise which they kept improving. Also in the absence of Generalists, the Specialists were able to bid on a variety of tasks and kept increasing their learning gain. In the 50-00-50 configuration, on account of the absence of the Specialists as teammates, the Generalists did not gain much in their learning from each task. However, the Apprentices on the other hand kept learning from Generalists and showed a larger learning gain consistently over time until the Apprentices gained enough expertise to render learning non-effective towards the end of the simulation. Another observation is that the learning gain of the Specialists was higher in the 00-50-50 environment as compared to the learning gain of the Generalists in the 50-00-50 environment. Because the Specialists were good at very few capabilities as compared to the Generalists who were good at a wider variety of capabilities, the Specialists had more diverse "rooms" to improve as compared to the Generalists, even just learning from the other Apprentices the capabilities in which the Specialists had very low level of expertise. This makes the learning gain of the Specialists higher than that of the Generalists.

4.6 Investigation 3: Impacts of Agent and Task Openness and Agent/Task Diversity

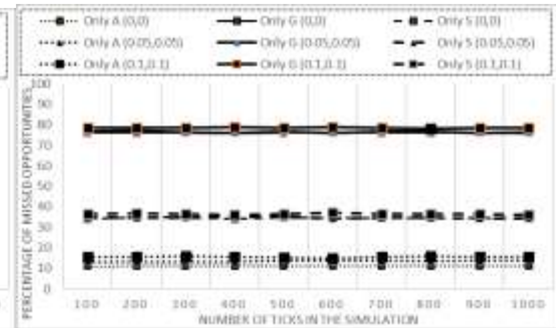
In this section, we investigate the impacts of Agent and Task Openness and agent diversity on the percentage of tasks auctioned off and the learning gains of the agents. Since this is the first time we put forth a step towards understanding the impacts of combining openness and diversity, instead of looking at both agent diversity and task diversity together in an open environment, we investigate these two aspects of diversity separately. Note that as alluded earlier in Chapter 1, we will address the impacts of all these aspects of openness and diversity in Chapter 5.

4.6.1 Impacts on average percentage of missed opportunities

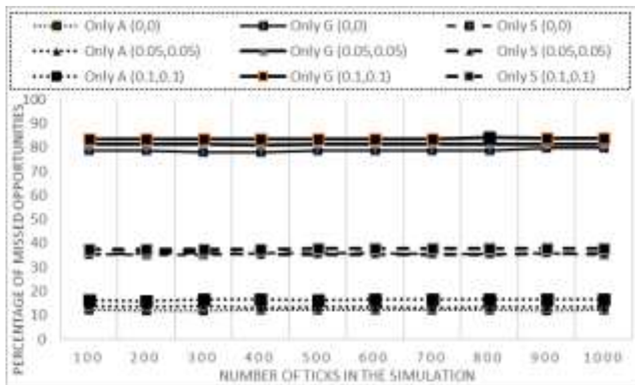
In this Section, we investigate whether and how both openness and task diversity have an impact on the average percentage of missed opportunities. As seen from Figure 4.16, the average percentages of missed opportunities remained the same throughout the 1000 ticks even with an increase in the level of TD but in the absence of openness. One reason for this is that in a closed environment, there are many permutations of tasks not auctioned off and agents that won a bid but the task wasn't auctioned off. This makes it difficult for the agents to bid for a probable task that will be auctioned off. To further investigate into this in the presence of openness, in Figure 4.16. we carried out configurations for all three TD (Low, Medium, and High), with Only A, Only G, and Only S with AO = TO = 0.05 and 0.1, respectively. Note, in the figure legend, for example, "Only S (0.05, 0.05)" means Only Specialists with AO = TO = 0.05.



(a) TD = Low



(b) TD = Medium



(c) TD = High

Figure 4.16. Average (over 100 runs) Percentage of missed opportunities for all three TD (Low, Medium, and High) configurations with Only A, Only G, and Only S, with AO = TO = 0, 0.05, and 0.1, respectively, $p < 0.0001$.

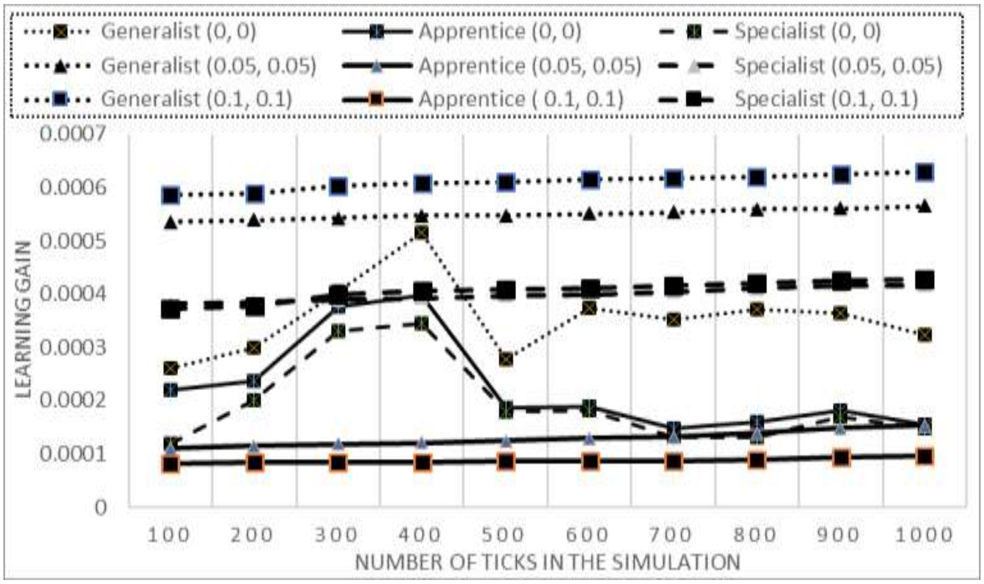
It can be seen that the average percentage of missed opportunities increased for all the agent types as the level of Task Diversity increased along with the task and agent openness. Openness introduces uncertainty to the task-diverse environment causing even more missed opportunities for the agents. Even if certain teammates were comfortable working with each other on a certain task and running it to completion, openness could cause either of the teammates or the task that they had worked on to be removed from the environment, which causes an imbalance in this harmony that they had established and that led to more missed opportunities. New tasks and agents kept entering the environment as well which prompted the agents to keep *remodeling* their probabilities

($p_{auctioned}$ of Equation 3.8) of bidding and getting tasks assigned, in the process of which they might miss out on tasks.

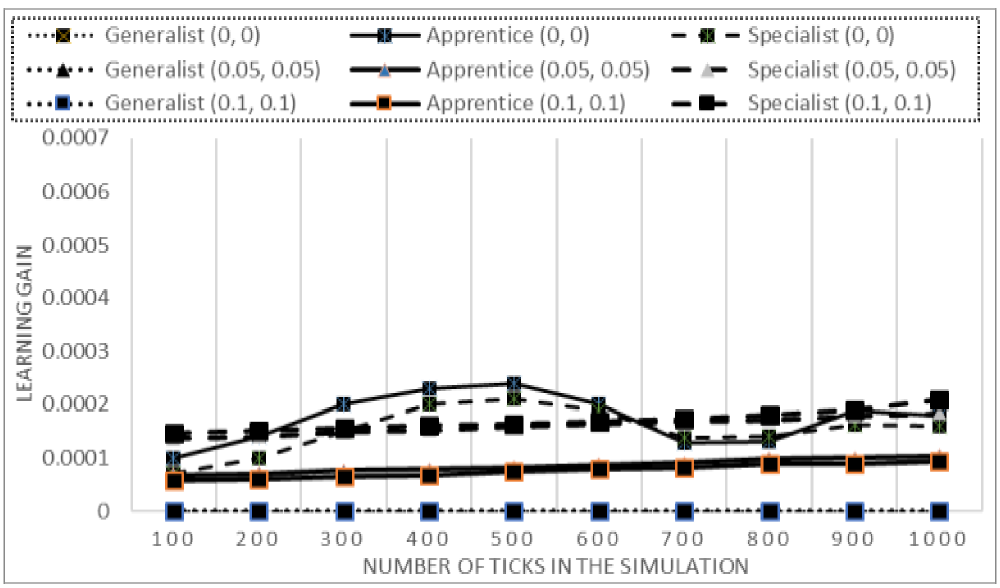
From Figure 4.16, the average percentages of missed opportunities increased with the level of task diversity as well as openness. As seen earlier in Section 4.4 as well, this happened since all the agents were in a long chase of finding teammates; that is why they kept bidding for different tasks at each auction round rather than all the agents coming together. Openness complicated this chase further by introducing uncertainty about the presence of teammates and tasks which caused the agents to constantly remodel their probabilities of winning a bid cast.

4.6.2 Impacts on Learning

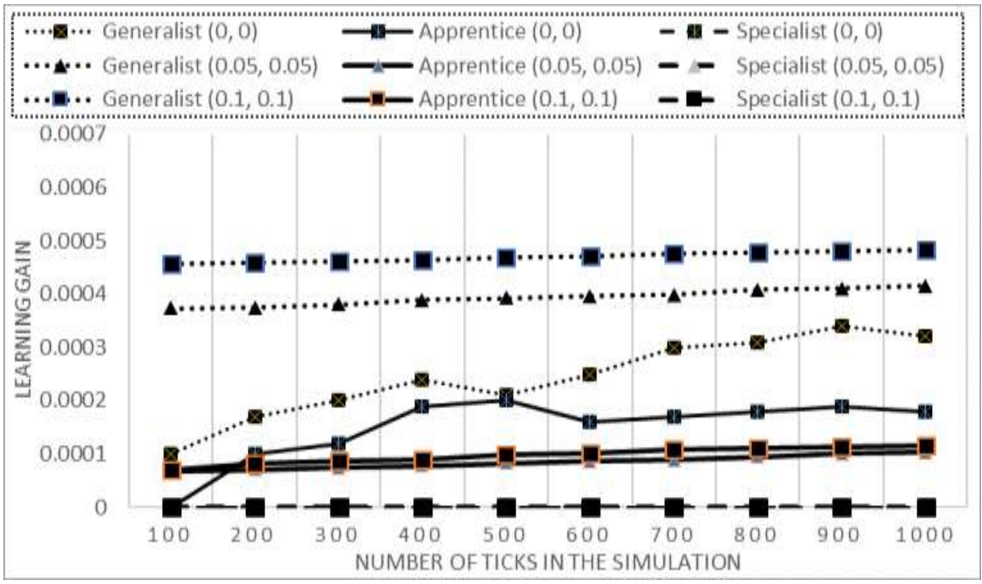
Here we investigate if agent and task openness have an impact on agent's learning. Figure 4.17 (a)-(c) shows the impact of (AO = 0, TO = 0), (AO = 0.05, TO = 0.05) and (AO = 0.1, TO = 0.1) on the learning gain per tick of agents for all three AD configurations and TD = Medium, respectively. Note, in the figure legend, for example, "Specialist (0.05, 0.05)" means the configuration with only Specialists with AO = TO = 0.05. Again, we did not consider both types of diversity here, and only looked at how agent diversity and openness interact to impact agent learning. And we will present a detailed analysis of the learning gains in an agent- and task -open, and agent- and task-diverse environment in Chapter 5.



(a) AD 25-25-50



(b) AD 00-50-50

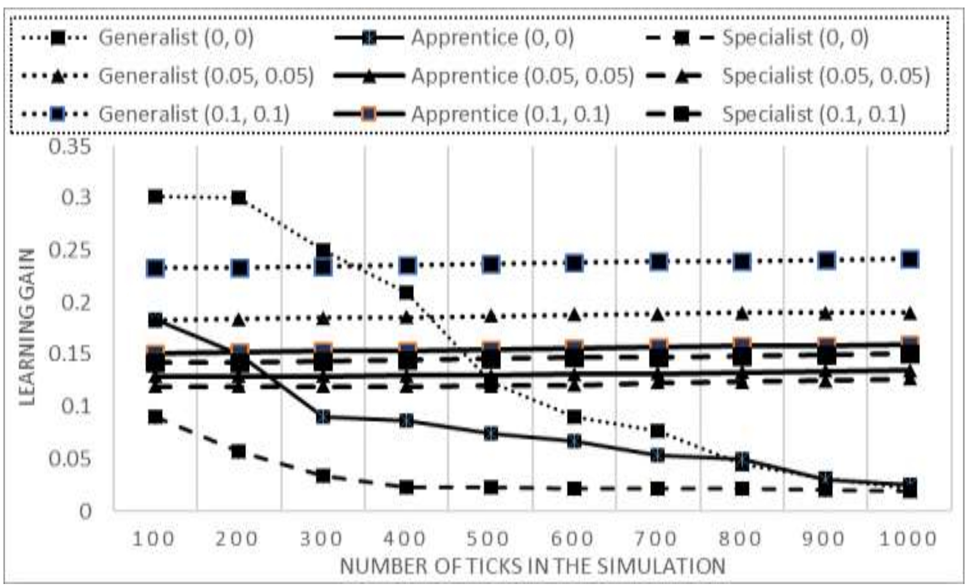


(c) AD 50-00-50

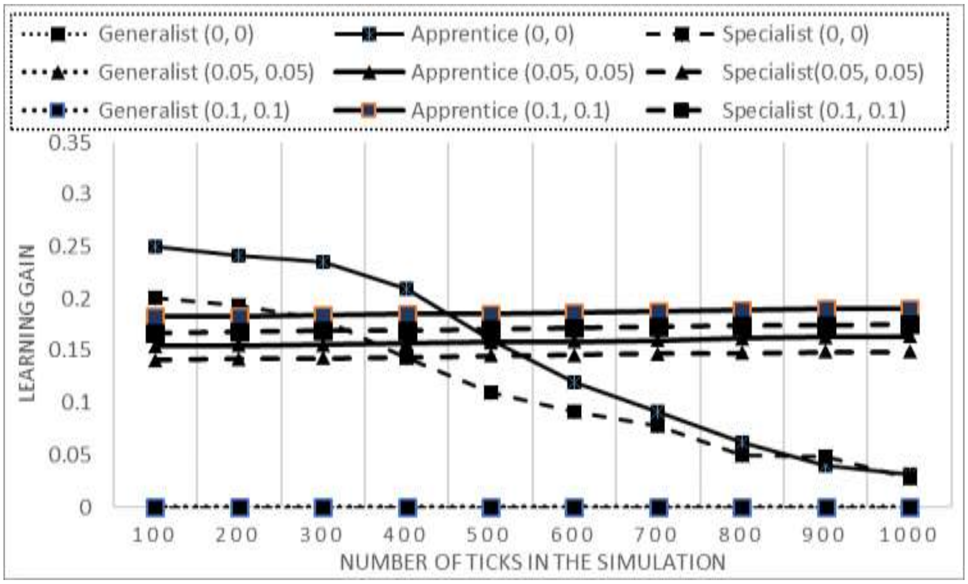
Figure 4.17. Average (over 100 runs) learning gain per tick for all three AD configurations, with TD = Medium, AO = TO = 0, 0.05, and 0.1, $p < 0.0001$.

From Figure 4.17 it can be observed that as agent and task openness increased from 0 to 0.05 to 0.1, the average learning gains for all three agent types increased for all three AD configurations. Thus, this shows that openness does have an effect on the learning gains of the agents. In fact, it can also be seen that when TO and AO was non-zero, the learning curves did not converge. We suspect that since agents have some capability they keep improving by observing new agents or working on new tasks, their knowledge never reaches a stagnant still and they keep learning more and evolving. To confirm this suspicion, we further investigated to find if agents keep sharpening their skills by working on new tasks or teaming up with new agents. We found that at 86.7% of the time, agents kept on improving their skills because they worked on tasks they had not before or they teamed up with agents with capabilities they had not seen before. Thus, openness creates a balance between old and new capabilities and presents agents with an opportunity to keep learning more skills and be more beneficial to the environment. This also explains the flat learning curves for the non-zero openness.

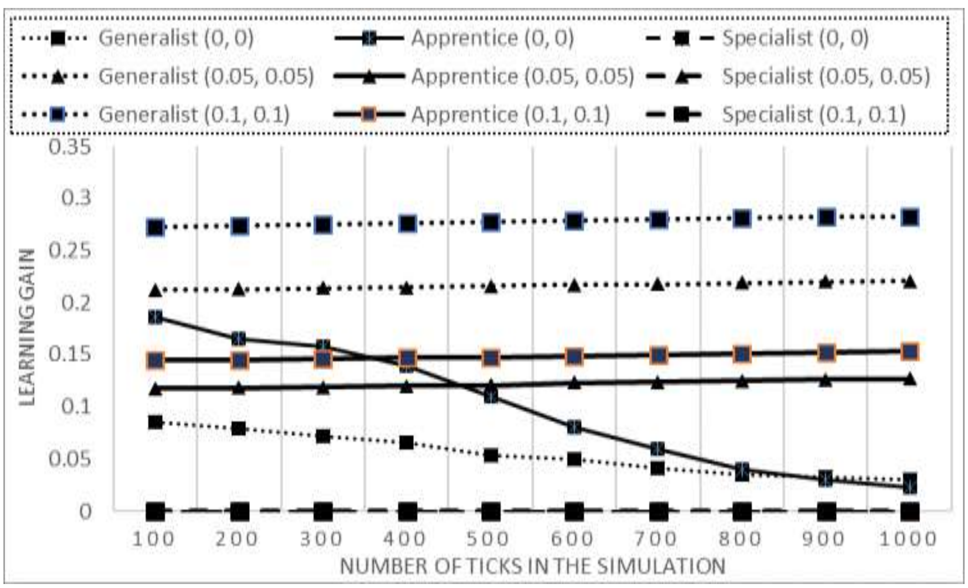
Figure 4.18 represents the average learning gain per task for every 100th tick of the 1000 ticks, for (AO = 0, TO = 0), (AO = 0.05, TO = 0.05) and (AO = 0.1, TO = 0.1) on the learning gain per tick of agents for all three AD configurations and TD = Medium, respectively. Note, in the figure legend, for example, "Specialist (0.05, 0.05)" means Specialists with AO = TO = 0.05.



(a) AD 25-25-50



(b) AD 00-50-50



(c) AD 50-00-50

Figure 4.18. Average (over 100 runs) learning Gain per task for all three AD (25-25-50, 00-50-50, and 50-00-50) configurations, with TD = Medium, with AO = TO = 0, 0.05, and 0.1, $p < 0.0001$.

It can be seen from Figure 4.18 that an increase in AO and TO caused the learning gain per task to increase as well. This is because on account of new tasks coming in and new agents coming in or old ones leaving, agents always have something new to learn by doing tasks. This causes an increase in the learning gain with increasing AO and TO.

It can be observed that for non-zero openness (AO = TO = 0.05 or 0.1) the learning gain curves showed a slight increasing trend—meaning that the agents were learning throughout the entire duration. This is because openness presented the agents with an opportunity to work on new tasks and with new teammates. This opportunity helped the agents improve their capabilities consistently. On the contrary, in the case of no openness (AO = TO = 0), agents kept working with the same set of teammates with the same set of tasks. As a result, they reached a point beyond which they stopped learning anything new and the learning curves thus showed a downward trend, as previously discussed in Section 4.3.2.

As seen earlier in Section 4.3.2 under Learning gains over time, Generalists had higher average learning gains than the other agent types. A similar trend was also observed from Figure 4.18 (a) and (c), the Generalists had higher average learning gains that their counterparts. This is because the Generalists had a decent level of expertise at a wider variety of capabilities which helped them be fit to bid and win a wider variety of tasks as compared to the other types of agents. This in return presented the Generalists an opportunity to keep learning something new with each task by doing and observation. It can also be seen that in the absence of the Generalists, the Apprentices and Specialists saw a rise in their learning gains. Also, it can be seen that in the absence of the Specialists, the Generalists saw a rise in the learning gain since they now get assigned to tasks which were assigned to Specialists had they been present. Thus, we see evidently that Agent and Task openness along with agent diversity has an impact on the learning gains of the agents. Introduction of openness to an agent diverse environment caused positive impacts on the learning gains of agents. This positive impact was attributed to the continuously increasing levels of the learning gains in the presence of openness in an agent diverse environment.

4.6.3 Discussions

The key observation and findings were:

(1) From Section 4.6.1 we realized that, the average percentages of missed opportunities increased with the level of task diversity as well as openness. This happened since all the agents were in a long chase of finding teammates; that is why they kept bidding for different tasks at each auction round rather than all the agents coming together. Openness complicated this chase further by introducing uncertainty about the presence of teammates and tasks which caused the agents to constantly remodel their probabilities of winning a bid cast.

(2) Section 4.6.2 helped us realize that with the increase in the level of openness and agent diversity, the learning gains showed an increase as well. This is because on account of new tasks coming in and new agents coming in, agents always have something new to learn by doing tasks and observing agents. We also observed that Generalists had higher average learning gains than the other agent types. This is because the Generalists had a decent level of expertise at a wider variety of capabilities which helped them be fit to bid and win a wider variety of tasks as compared to the other types of agents.

4.7 Summary

In this section, we present a brief summary of the findings of our investigations and point out the weaknesses of our results.

The findings of our investigations helped us realize that both diversity and openness have an impact on the team formation and how agents learn to form these teams. Furthermore, we breakdown the key findings of our investigations:

Impacts on agent performance. From the analyses of diversity and openness on average rewards per tick (Section 4.2) we see that *the average rewards earned per tick by the agents increases with the increase in the level of agent- and task-, diversity and openness*. However, upon observing the rate of rise of the average rewards per tick, we realize that the *introduction of openness to an agent and task diverse environment, decreases the predictability of the impacts of both diversity and openness together on agent performance*. This is based on the observation of the rates of rise of the average rewards earned per tick not following a linear pattern when openness was introduced along with diversity. On the other hand, in case of a diverse environment alone, these rates of rise of the average rewards per tick did follow a consistent pattern, making the impacts of diversity alone quite predictable.

AD versus TD on average percentage of tasks auctioned off. From Investigations I, II, and III, we now know that the average percentage of tasks auctioned off (or the number of task completed) increases with the level of agent and task diversity, respectively. Since we have established a direct correlation between average rewards per tick and the average percentage of tasks auctioned off (Section 4.2), Investigations I, II, and III helped us realize that, *a diverse environment does prove beneficial for the agent performance*. To be able to observe which diversity has a greater impact on the agent performance, we analyze the slopes of the average percentage of tasks auctioned off under agent and task diversity, separately (Section 4.3). The *p*-values across these slopes prove that both AD and TD are statistically significantly different. The slopes for AD are higher than those for TD, thus showing that *AD has a greater impact on the agent performance than does TD*.

Evolution (Learning gains). Furthermore, from Investigations I, III, and Section 4.5, we also understand that *with the increase in the level of agent and task diversity the average learning gains increases as well*. To be able to analyze which of AD or TD has a greater impact on the evolution/learning gains of the agent, again, we look at the slopes for the average learning gains under each. The *p*-values prove that both the learning gains under AD and TD are statistically significantly different. On comparing the slopes of the average learning gains per tick we realize that *the impacts of AD are greater than those of TD on the evolution of the agents*.

Average percentage of missed opportunities. *With the increase in the level of task diversity and openness, the average percentage of missed opportunities increases as well* (as shown in Investigation III). This is because agents end up in a long chase of chasing each other around to find teammates, rather than focusing on common tasks. To overcome this chase in an open and diverse environment, we propose a refined approach in Chapter 5, with the goal of helping agents choose common tasks more frequently to bid on and find the needed number of teammates, rather than chasing each other around.

There are a few weaknesses of our results. First, we did not calculate the standard deviations. The standard deviations would have helped us understand the distribution of various metrics in our investigations. However, we failed to keep the raw data due to storage issues and neglected to compute the standard deviations before deleting them. Second, we studied only some AO-TO combinations ((AO, TO) = (0,0), (0.05, 0.05), and (0.1, 0.1)) instead of all the 9x9 combinations. This was done to save time as running all 9x9 combinations would have taken much more time in both simulations and subsequent analyses. We thought that by only looking at the above three combinations, we would have sufficient representation and insights. Having now derived insights from the data, it would make sense to continue with all combinations in the future to fill in the gap and to further confirm the above findings.

Chapter 5

$p_{auctioned+}$ ANALYSIS

In this chapter, we compare $p_{auctioned+}$ and $p_{auctioned}$ in terms of key performance metrics: average rewards per tick, average rewards per task, average percentage of tasks completed (i.e., auctioned off), average learning gains per tick, and average learning gains per task. *Note, average throughout this chapter refers to the average over the 100 runs for all the configurations.* Recall that $p_{auctioned+}$ is designed to help maximizing utility in an agent- and task-diverse environment, specifically by reducing or preventing agents from "chasing each other around" while submitting bids to successfully complete tasks. We conducted a series of investigations to help us test whether and why $p_{auctioned+}$ performs better than $p_{auctioned}$.

5.1 Comparisons between $p_{auctioned+}$ and $p_{auctioned}$

In this section, we compare $p_{auctioned+}$ and $p_{auctioned}$ in terms of key performance metrics: average rewards per tick, average rewards per task, average percentage of tasks completed (i.e., auctioned off), average learning gains per tick, and average learning gains per task. Tables 5.1-5.5 show the averages for rewards per tick, average rewards per task, average percentage of tasks completed (i.e., auctioned off), average learning gains per tick, and average learning gains per task, for $p_{auctioned}$ and $p_{auctioned+}$, respectively. The configurations used for these tables are: AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, Medium, and High.

		AD = 25-25-50			AD = 00-50-50			AD = 50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	0.0035	0.0046	0.0070	0.0014	0.0025	0.0051	0.0024	0.0041	0.0065
	0.05, 0.05	0.0066	0.0067	0.0094	0.0029	0.0038	0.0069	0.0046	0.0053	0.0074
	0.1, 0.1	0.0068	0.0076	0.0095	0.0037	0.0056	0.0074	0.0049	0.0070	0.0086

(a) $p_{auctioned}$

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	0.0049	0.0063	0.0084	0.0040	0.0058	0.0078	0.0043	0.0062	0.0081
	0.05, 0.05	0.0068	0.0069	0.0096	0.0052	0.0056	0.0087	0.0059	0.0065	0.0092
	0.1, 0.1	0.0074	0.0079	0.0098	0.0061	0.0068	0.0091	0.0068	0.0071	0.0094

(b) $p_{auctioned+}$

Table 5.1. Average rewards earned per tick, for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, Medium (Med), and High, (a) $p_{auctioned}$, and (b) $p_{auctioned+}$, standard error = 0.0003.

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	0.0396	0.0426	0.0454	0.0333	0.0400	0.0402	0.0390	0.0423	0.0431
	0.05, 0.05	0.0420	0.0532	0.0553	0.0401	0.0438	0.0499	0.0411	0.0520	0.0546
	0.1, 0.1	0.0545	0.0565	0.0630	0.0459	0.0487	0.0606	0.0500	0.0519	0.0622

(a) $p_{auctioned}$

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	0.0429	0.0465	0.0468	0.0397	0.0412	0.0422	0.0402	0.0453	0.0440
	0.05, 0.05	0.0510	0.0547	0.0564	0.0454	0.0492	0.0507	0.0499	0.0534	0.0563
	0.1, 0.1	0.0547	0.0581	0.0697	0.0529	0.0560	0.0669	0.0538	0.0567	0.0676

(b) $p_{auctioned+}$

Table 5.2. Average rewards earned per task, for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, Medium (Med), and High, (a) $p_{auctioned}$, and (b) standard error = 0.0003.

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	60.8548	65.0392	66.4827	50.5746	55.4833	59.9183	58.5869	60.1892	62.3828
	0.05, 0.05	63.2929	66.1943	68.9275	52.1832	58.8765	61.2282	60.0193	65.8832	66.1911
	0.1, 0.1	64.1939	70.5873	74.326	55.4883	60.1930	65.7865	61.1838	69.1921	72.1885

(a) $p_{auctioned}$

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	65.4320	66.5432	71.2000	53.4332	60.5838	63.3372	60.9273	62.5982	66.5921
	0.05, 0.05	69.0000	72.8922	74.5731	57.5890	68.1382	70.9737	65.6832	70.4881	73.5939

	0.1, 0.1	70.3221	76.4383	78.0098	59.9281	71.9982	74.5432	68.5929	75.0282	77.2478

(b) $p_{auctioned+}$

Table 5.3 Average percentage of tasks auctioned off for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, TD = Low, Medium (Med), and High respectively, (a) $p_{auctioned}$, and (b) standard error = 0.0004.

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	6.43E-04	7.29E-04	8.00E-04	3.28E-04	3.58E-04	4.67E-04	5.37E-04	6.37E-04	7.38E-04
	0.05, 0.05	7.68E-04	8.34E-04	8.57E-04	4.92E-04	5.92E-04	6.84E-04	6.38E-04	7.64E-04	8.14E-04
	0.1, 0.1	8.61E-04	8.92E-04	9.01E-04	5.38E-04	6.11E-04	7.18E-04	7.39E-04	8.49E-04	8.59E-04

(a) $p_{auctioned}$

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	7.22E-04	7.96E-04	8.03E-04	4.56E-04	5.38E-04	5.55E-04	6.39E-04	6.99E-04	7.99E-04
	0.05, 0.05	7.99E-04	8.55E-04	8.76E-04	4.98E-04	6.18E-04	6.98E-04	6.77E-04	7.99E-04	8.16E-04
	0.1, 0.1	8.92E-04	8.98E-04	9.11E-04	6.68E-04	7.38E-04	7.76E-04	8.32E-04	8.68E-04	9.04E-04

(b) $p_{auctioned+}$

Table 5.4 Average learning gains per tick for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, TD = Low, Medium (Med), and High respectively, (a) $p_{auctioned}$, and (b) standard error = 0.0002.

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	5.33E-03	6.00E-03	7.00E-03	5.03E-03	5.48E-03	6.34E-03	5.21E-03	5.76E-03	6.93E-03
	0.05, 0.05	6.45E-03	6.89E-03	7.90E-03	5.82E-03	5.85E-03	6.47E-03	6.01E-03	6.40E-03	6.67E-03
	0.1, 0.1	7.38E-03	7.48E-04	8.75E-03	6.03E-03	6.61E-03	7.08E-03	6.55E-03	6.98E-03	7.44E-03

(a) $p_{auctioned}$

		AD = 25-25-50			AD = 00-50-50			AD =50-00-50		
	TD	*Low*	*Med*	*High*	*Low*	*Med*	*High*	*Low*	*Med*	*High*
AO, TO	0, 0	6.41E-03	6.89E-03	7.03E-03	6.33E-03	6.49E-03	7.00E-03	6.39E-03	6.55E-03	7.02E-03
	0.05, 0.05	6.55E-03	6.92E-03	8.02E-03	5.96E-03	6.01E-03	7.11E-03	6.48E-03	6.59E-03	7.53E-03
	0.1, 0.1	7.83E-03	7.99E-03	8.88E-03	6.44E-03	6.69E-03	7.66E-03	7.77E-03	7.86E-03	7.89E-03

(b) $p_{auctioned+}$

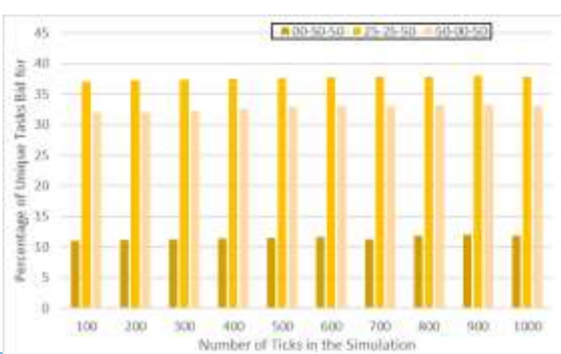
Table 5.5. Average learning gains per task for AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, TD = Low, Medium (Med), and High respectively, (a) $p_{auctioned}$, and (b) $p_{auctioned+}$ standard error = 0.0002.

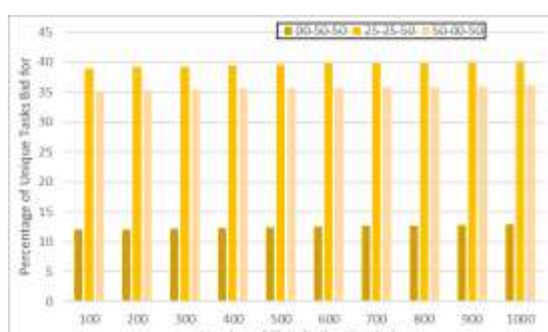
As observed from the Tables 5.1-5.5, $p_{auctioned+}$ outperformed $p_{auctioned}$ in all performance metrics, in all configurations.

5.2. Reasoning for $p_{auctioned+}$ outperforming $p_{auctioned}$

In this section, we present the reasoning for why $p_{auctioned+}$ performed better than $p_{auctioned}$ as seen from Section 5.1. As seen from Chapter Methodology, $p_{auctioned+}$ was designed such that it could reduce the agents chasing each other around. We intuit that the reason why $p_{auctioned+}$ outperformed $p_{auctioned}$ was because with $p_{auctioned+}$ the agents were realizing to stop chasing each other and to focus on common tasks which would more likely get the needed number of teammates bids for the task to be auctioned off. Thus, this reduction in the chase around led to $p_{auctioned+}$ performing better than $p_{auctioned}$.

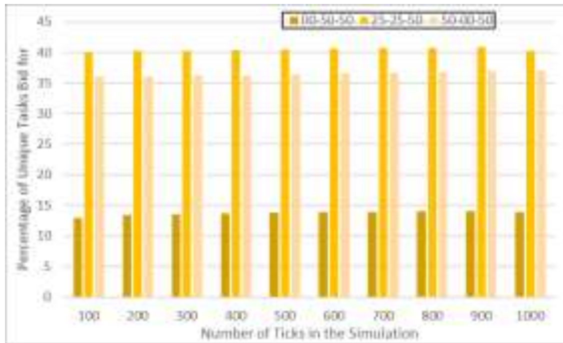
To confirm this intuition, in the $p_{auctioned+}$ solution, if more agents did start coming together to work on tasks rather than chasing each other, then this would hint that the number of different tasks for which agents submitted their bids with the $p_{auctioned+}$ should be smaller than that in the $p_{auctioned}$ solution, since agents would be coming together more often to work on common tasks. To prove this, we take a look at the average percentage of unique tasks that agents bid for under both $p_{auctioned}$ and $p_{auctioned+}$. Figures 5.1 and 5.2 present the average percentage of unique tasks bid for $p_{auctioned}$ and $p_{auctioned+}$, respectively with AO = TO = 0, 0.05, and 0.1, AD = 25-25-50, 00-50-50, and 50-00-50, and TD = Low, Medium, and High, respectively, over time, after every 100th tick in the simulation.



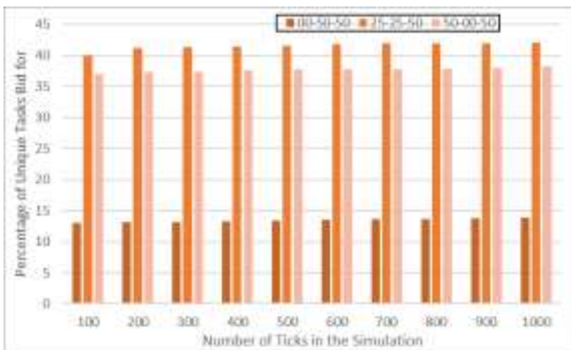


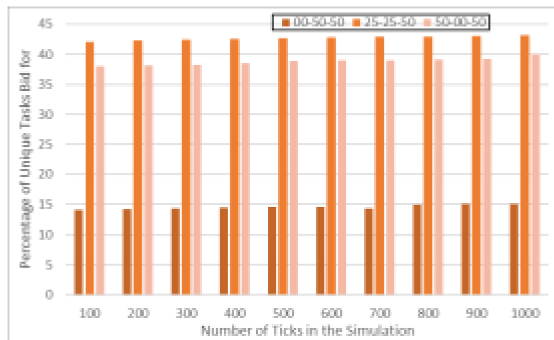
(a) TD = Low

(b) TD = Medium



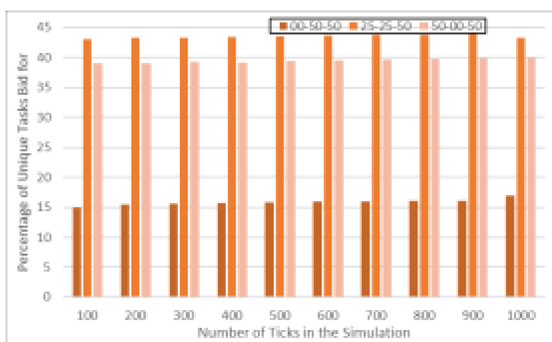
(c) TD = High



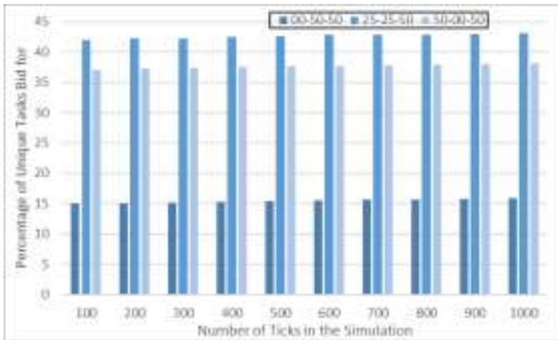


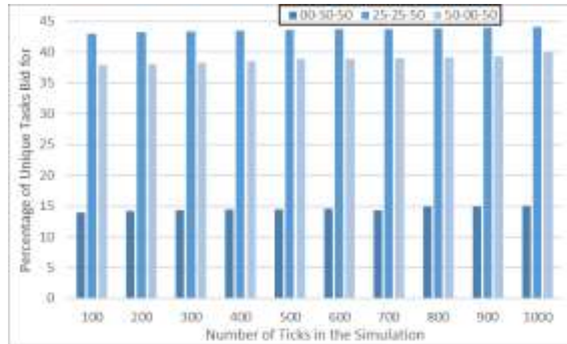
(d) TD = Low (0.05, 0.05)

(e) TD = Medium (0.05,0.05)

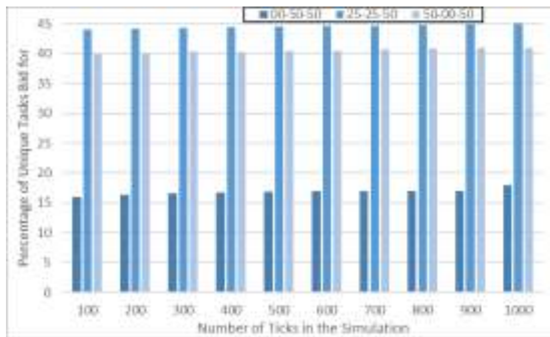


(f) TD = High (0.05,0.05)



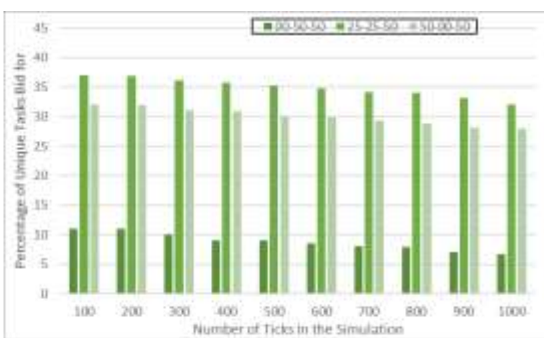


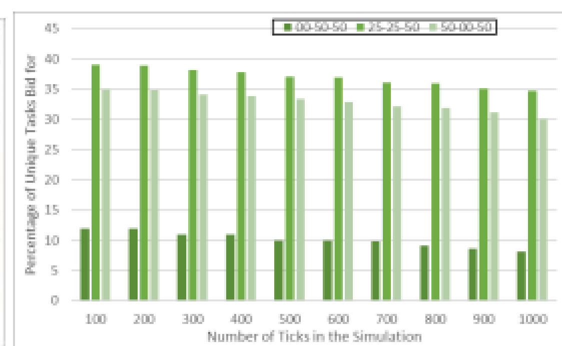
(g) TD = Low (0.1,0.1) (h) TD = Medium (0.1,0.1)



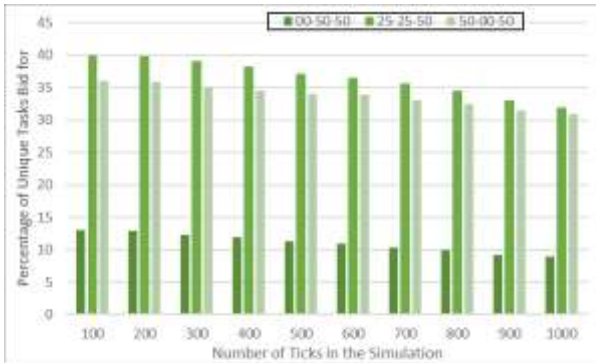
(i) TD = High (0.1,0.1)

Figure 5.1 Average percentage of unique tasks bid for TD = Low, Medium, and High, with AD = 25-25-50, 00-50-50, and 50-00-50, AO = TO = 0, 0.05, and 0.1, for $p_{auctioned}$, $p < 0.0001$.

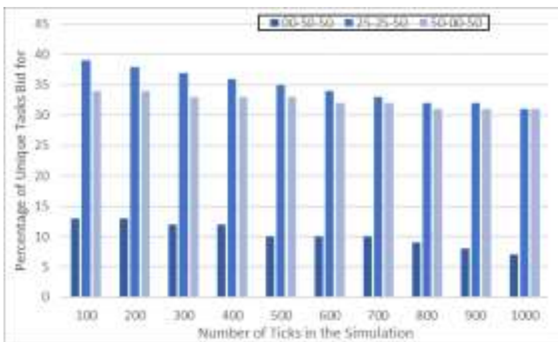




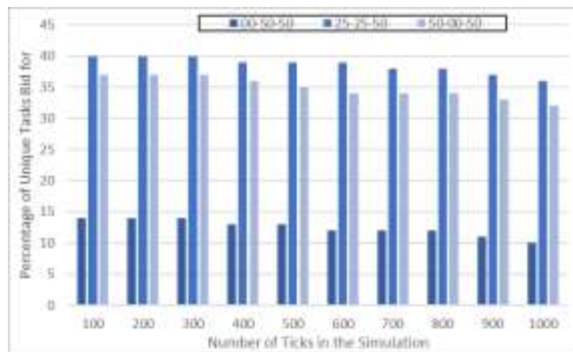
(a) TD = Low (0,0) (b) TD = Medium (0,0)



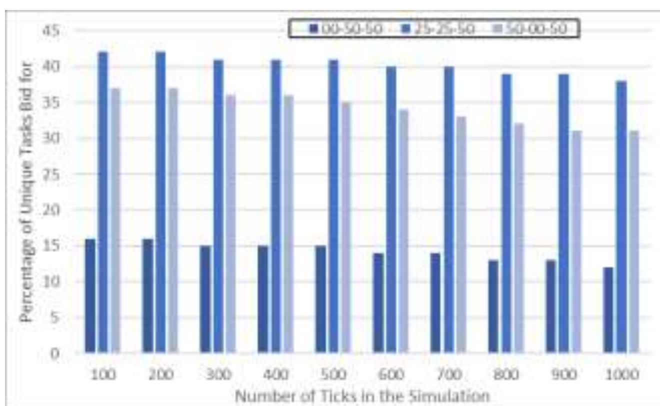
(c) TD = High (0,0)



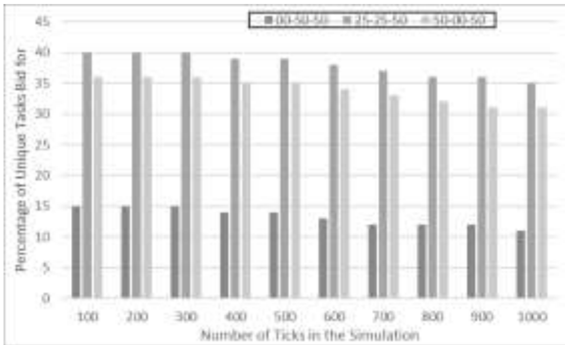
(d) TD = Low (0.05, 0.05)



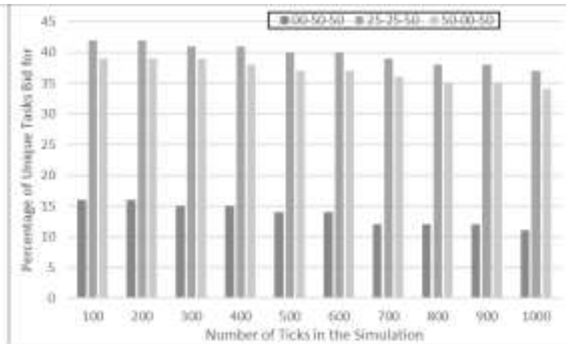
(e) TD = Medium (0.05,0.05)



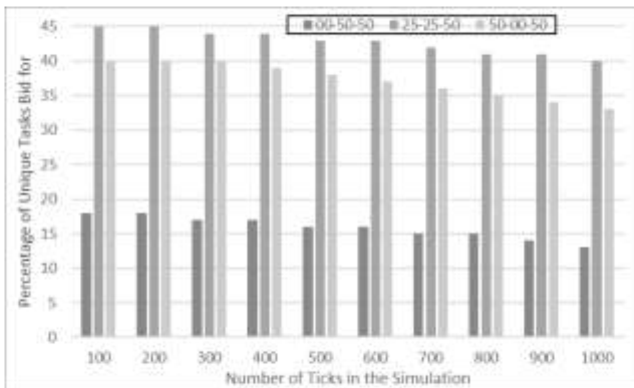
(f) TD = High (0.05,0.05)



(g) TD = Low (0.1,0.1)



(h) TD = Medium (0.1,0.1)



(i) TD = High (0.1,0.1)

Figure 5.2 Average percentage of unique tasks bid for TD = Low, Medium, and High, with AD = 25-25-50, 00-50-50, and 50-00-50, AO = TO = 0, 0.05, and 0.1, for $p_{auctioned+}$, $p < 0.0001$.

As seen from Figures 5.1 and 5.2, $p_{auctioned+}$ did reduce the average percentage of unique tasks that agents bid for. We see that $p_{auctioned+}$ guided the agents in the direction of tasks which would have a higher chance of being auctioned off because of presence of sufficient number of teammates. As opposed to $p_{auctioned}$, which only guided agent with the information of a task being auctioned off or not irrespective of the number of teammates that might be present. By not providing the teammate posts filled information, the agents would not be able to realize how close a task was to being auctioned off and would tend to keep bidding for different tasks with incomplete information and hence ending up getting fewer tasks auctioned off. As also seen from Figure 5.2, while the agents with $p_{auctioned+}$ did reduce the number of unique tasks and start bidding for common tasks, they did not repeatedly keep bidding for these common tasks, else the percentages would have gone down to be very close to 0. This is also a good thing that agents keep aiming for tasks which they probabilistically view to be assigned to them and also auctioned off. By working on a variety of tasks they also get to sharpen a wider variety of skills by working with different teammates. Agents tend to not bid for the same task throughout since as seen from Chapter Methodology, we see that the probabilistic modelling depends on similar tasks. Hence, at the start of every bid the agents calculate their chances of winning and getting a task auctioned off for a number similar tasks and not just one task which helps them choose from a number of task options.

5.3 Summary

In Section 5.1, we carried out analyses to test whether $p_{auctionted+}$ performed better than $p_{auctioned}$. To compare, we carried out analyses on the average rewards per tick, average rewards per task, average percentage of tasks auctioned off, average learning gains per tick, and average learning gains per task. The analyses showed that $p_{auctioned+}$ outperformed $p_{auctioned}$. This superior performance was because of agents realizing to bid on common tasks rather than chasing each other around. This reduction in the chase was further confirmed by analyzing the unique tasks agents were bidding for. Analyzing the unique tasks agents bid for showed that $p_{auctioned+}$ did reduce the average percentage of unique tasks agents bid for by helping them reduce the chase around.

Note, to further analyze how $p_{auctioned+}$ is better than $p_{auctioned}$, we carried out time-based analyses in Appendix A, on the average rewards per tick, average rewards per task, average percentage of tasks auctioned off, average learning gain per tick, and average learning gain per task for $p_{auctioned}$ and $p_{auctioned+}$ respectively. These comparison results helped us gain additional insights as to how $p_{auctioned+}$ performed better over time. We see that this reduction in the agent chase led to more tasks auctioned off timely and in return to higher rewards and learning gains over time, thus leading to better agent performance.

Having established that $p_{auctioned+}$ does perform better than $p_{auctioned}$, the next question is the reason behind it being better. Recall, our goal is to keep the probabilistic modelling (Chapter Methodology) as accurate as possible with no/minimal pre-coordination (Chapters Introduction and Methodology). By providing the information about the unfilled teammate posts we still keep the pre-coordination minimum, since this is the information the agents deserve to know as it is the information which is a part of the result of the tasks the agent bid for. But is sharing of this information of the unfilled teammate posts the main reason behind $p_{auctioned+}$ doing better? Or is giving the information about the unfilled posts behaving similarly to a self-fulfilling prophecy for the agents? Having received the information about the number of unfilled teammate posts, the agents take the hint to stay put in a hope that the wandering agents will come find this task. Is it possible that this action of the agent to stay put lead to the better performance by $p_{auctioned+}$? As a part of the future work, we wish to dive deeper to investigate what was contributing to the better performance of $p_{auctioned+}$, (1) Including the information about unfilled teammate posts in $p_{auctioned+}$, or/and (2) Agents taking the hint to stay put and these actions causing the $p_{auctioned+}$ to get better.

Chapter 6

IMPLEMENTATION

In this chapter, we describe the implementation details of the model Chen (2017) which we extended to incorporate diversity as well. The environment is based off a simulator called Multi Agent Ad-Hoc Team Formation Simulator+ (MAAHTFormS+), which is extended from Chen (2017). However, since the original model implementation Chen (2017) was not designed with object oriented concepts, most of the codebase had to be re-organized under different parent and child classes. This re-organization involved a lot of code re-write as well. Also, the original model did not handle invoking of objects and inheritance of classes in a straightforward manner which lead to further re-writing of the codebase. The original implementation did not calculate a number of metrics used in our research, such as, the number of unfilled teammate posts, number of unique tasks agents bid for (Chapter Methodology), etc. However, since the analyses in our research required these metrics, we had to add more code to be able to compute these metrics. To be able to deal with the results of these new metrics we had to code new post-processing scripts as well. Thus, along with basing the original model into the object-oriented structure, we also had to add many additional modules.

6.1 Programming Language

We chose Java as the primary programming language for our environment, for it being robust, object oriented and is platform-independent. Java helped us organize our codebase into modular units or classes. So that the whole code wasn't dependent on a single function or file, but each functionality had its own unique class and file. This also helps with the extensibility because if there is only a specific functionality that needs to be updated then we do not re-code the entire codebase but just the class corresponding to that functionality.

The post-processing of the data and the plotting of graphs were done in Python. Python is an extensive support for libraries and third-party modules which makes post-processing and graph plotting seamless. The built-in and dictionary based data structures made the post-processing on a lot of data really easy. Python has its own unit testing framework along with its strong text processing capabilities which makes it high in speed and productivity. This high-speed processing helped us post-process our results faster.

6.2 Integrated Development Environment

Eclipse was our primary Integrated Development Environment (IDE). We picked Eclipse over other IDE environments since it is open source and free. Code completion comes in handy since it saves the time of digging through the documentation. Syntax checking helps get rid of Syntax errors as and how the code is written rather than waiting till the compilation time. Refactoring was also of great help since there might come instances where a lot of renaming might be required for such a huge codebase.

6.3 Codebase build-up

Figure 6.1 shows the three major modules of the codebase.

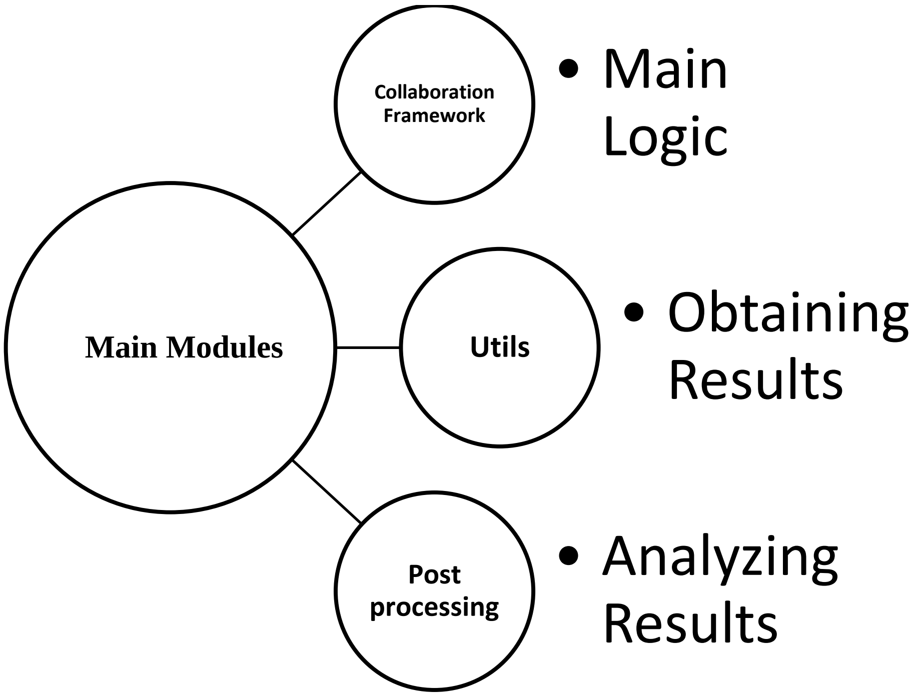


Figure 6.1. Overview of the main modules of the codebase.

6.3.1 Collaboration

This module includes the entire logic of the working on the environment. This module is further divided into object classes for each of the individual functionalities. The main division of this parent is as follows:

(1) Agent.java

This class is responsible for all the properties and activities that are related to agents, for example, agents viewing the blackboard for the tasks available for the auction, submitting bids, reading the assigned tasks, executing the tasks, etc. Recall, each agent has a type associated with it, (1) Apprentice, (2) Generalist, or

(3) Specialist (Chapter Methodology). Depending on the agent type, this class is responsible for assigning the capability vector with the corresponding level of expertise (Chapters Methodology and Introduction) to the agents.

(2) Blackboard.java

The primary function of this class is the functionality of the publish-subscribe system Wooldridge, (2009) and the central auctioneer. It is the primary medium of communication between the agents and the auctioneer. This parent class has modules for posting tasks, accepting bids, and posting winning agents and their corresponding tasks. The flowchart of the role of the central auctioneer is as shown in Figure 6.2 (described in detail in Chapter Methodology):

Figure 6.2. Central Auctioneer Working Flowchart.

(3) Debug.java

This class is for the developers working on this codebase. Since the codebase is large (approximately 16,552 number of lines) and it cannot be possible to keep extending more functionalities without debugging at regular intervals, we have added this class. Developers can add debug statements at breakpoints of their choice in order to be able to analyze the working of the codebase in details. This class has Booleans for switching debugging on or off. When the debugging is switched on the terminal of the IDE presents nice debug statements which can help the developer not only analyze if the code is working as expected but also understand the working on the system in detail. When the Boolean is toggled off, these debug statements will not be displayed. This toggling off is useful when the code is running on Supercomputers since debug statements cause I/O operations which can slow down the processing of the experiments.

(4) Learning.java

This module deals with the probabilistic modeling (Chapters Introduction and Methodology) and updating the capabilities of the agents that they gained through learning by observation or/and doing.

(5) Parameters.java

This class deals with reading the parameters that define the configuration, Agent and Task Openness, Agent and Task Diversity, Number of ticks in the simulation, and number of agents from the user (these parameters are described in detail in Chapter Methodology).

(6) Environment.java

This class is responsible for creating the entire environment based on the parameters specified by the user. It is also primarily responsible for introducing new agents and/or tasks into the environment based on the AO and TO parameters.

(7) Results.java

Since the code base is huge and serves many functionalities, the number of results achieved are also high. At a given time, it may not be required to produce all the results related to all the functionalities. In order to add flexibility to the results we allow users to specify which results they want, related to which functionalities. This not only saves on the processing time but also on the space used and the amount of data that the user receives.

6.3.2 Utils

The primary function of this module is to create the command files for a variety of utilities like:

(1) Creating the AO TO Timer

This module keeps a check on whether it is time for either Agent or Task replacements to occur. These timers can take any decimal value between 0 and 1, and are specified by the user. These timers are implemented as the likelihood of each agent or task being present after each tick. At the end of each tick, a uniform random decimal number generator generates a decimal number between 0 and 1. If this decimal number is less than or equal to AO or TO, then the corresponding agent and/or task will leave and a new agent and/or task will enter the environment. A departing agent or task is replaced by the same type of agent or task. For example, if an Apprentice is removed from the environment, then another Apprentice is added to the environment. However, the capability vector associated with the new agent or task, is randomly generated all over again depending on its type.

(2) Creating Agent Types

Based on the Agent Diversity configurations supplied, the different percentages of agent types are created by this file. Note, as explained in Chapter Methodology, this parameter is specified as %Generalists-%Specialists-%Apprentices. And the percentages of these individual agent types should always add up to a 100%. For example, if the AD configuration is 25-25-50, then 25% Generalists, 25% Specialists, and 50% Apprentices are created.

(3) Creating CSV Files

All the results required to be output are compiled together in nicely formatted CSV files. These CSV files are later supplied to the Python Scripts for post processing. Depending on the results that are being generated the content of these CSV files changes as well. These files are very large in size on account of the different parameters and their values that we log. The log files are generally ~30MB in size each (the size can be greater than 30MB as well, depending on the results that are being logged), and there are 50 random seeds generally. We log 3x3 (AO, TO) combinations, (0, 0), (0.05, 0.05), and (0.1, 0.1). Thus, the total number of the result files combined for one run can become: 3x3x50 = 450 files. Making the total size of one run of results to be ~450*30 ~13.5GB. We normally ran 100 such runs for each configuration for a given set of analyses, as indicated in Chapters 4 and 5 where we discussed our investigations and results. We ran over 40 different configurations for our analyses, and each of this configuration was run for 100 times. On account of so much data being generated, we ran into storage problems fairly soon.

(4) Creating Slurm Files

We use Slurm to run our code on Clusters. Slurm is an open source cluster management and job scheduling system. This module helps us create the thousands of files that we need for running the experiments on clusters.

(5) Creating Task Types

This module is used for creating different types of tasks based on the Task Diversity parameter specified. (Recall that we specify the type of the task as Low, Medium, or High (Chapter Methodology).)

6.3.3 Postprocessing

All the postprocessing is done in Python 3.0. Depending on the experiments executed, there are different Python scripts for postprocessing. They are written crisply and have comments at each step along with a brief description at the start of the program. These well documented programs help the user understand the working of the scripts. Along with this good documentation, the variable and function names are intuitive enough to know what's happening at each step as well. All the post-processing scripts are independent of each other and hence extensibility should be easy. To run any of the scripts the only command-line argument needed is the path to the results folder one is looking to postprocess.

6.4 Specifications of the Cluster the Experiments were run on

We ran our experiments on Tusker (cluster name), it is a 106-node Production-mode LINUX cluster. Its processors are Opteron 6272 2.1GHz, 4 CPU/64 cores per node. RAM specifications are: 256 GB RAM per node, 2 Nodes with 512GB per node, and 1 Node with 1024GB per node.

6.5 Execution Steps of the Environment

Figure 6.3 presents a brief flowchart for the execution steps of the environment. The detailed explanation of each step of this flowchart has already been presented in Chapter Methodology.

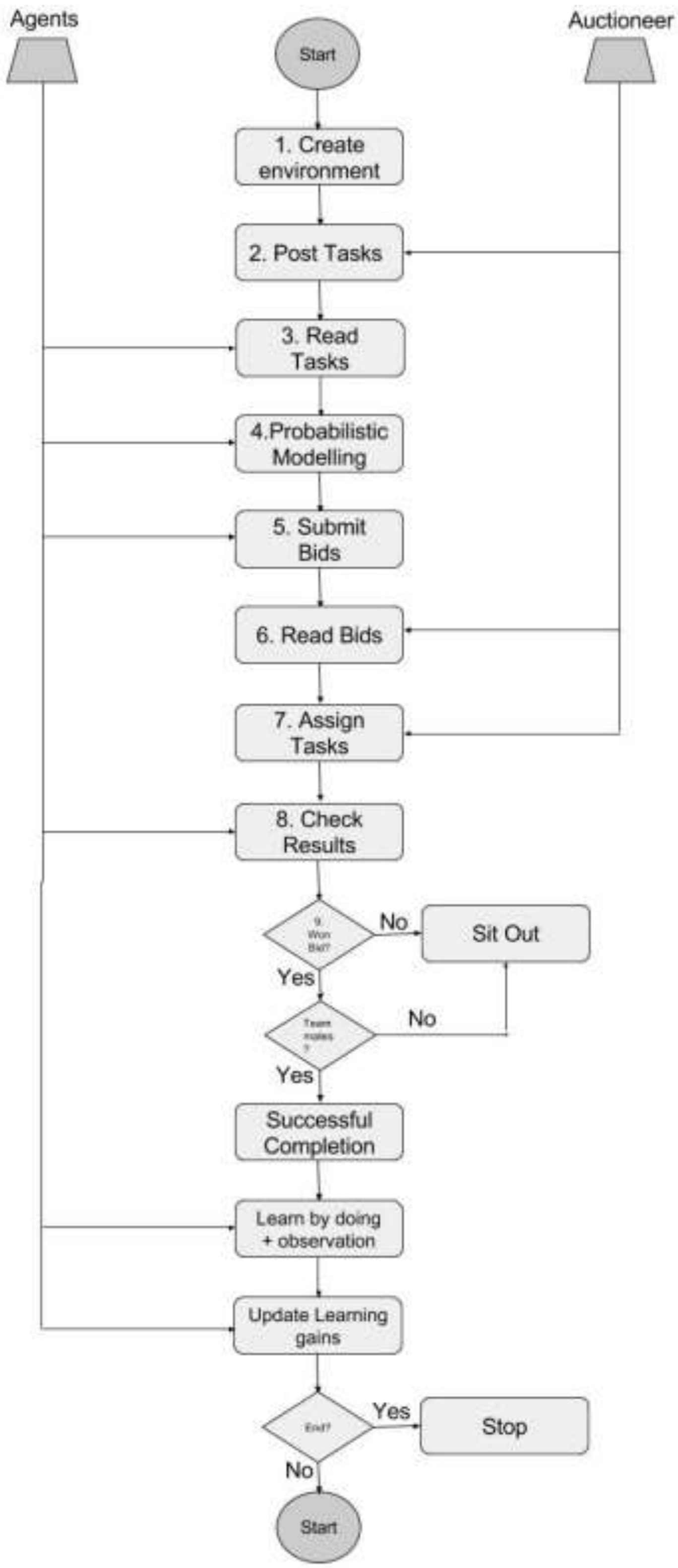


Figure 6.3. Flowchart of the working of the environment.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this Chapter, we compile together our contributions and key findings (Section 7.1). We also present our plans for the future work (Section 7.2).

7.1 Conclusions

In this section, we present the contributions of our research and the key findings of the investigations carried out. The primary goal of this thesis focused on analyzing the impacts of agent and task openness and diversity on the robustness and dynamics of team formation and how the agents learnt to form these teams. For this, several investigations were conducted with different permutations and combinations of diversity and openness to analyze at fine levels what the impacts of these could be on team formation. ***The findings of our investigations helped us realize that both diversity and openness have an impact on the team formation and how agents learn to form these teams***. After having deeply studied the findings of our investigations, we identified several key relations between diversity and openness: (1) Introduction of openness to an agent and task diverse environment decreased the predictability of the impacts of both diversity and openness together on agent performance, (2) A diverse environment did prove beneficial for the agent performance, (3) With the increase in the level of agent and task diversity the average learning gains increased as well, (4) Agent diversity had a greater impact on agent performance and the learning gains than task diversity did, and (4) With the increase in the level of diversity and openness, the average percentage of missed opportunities increased as well.

The next contribution of this thesis was to analyze why the average percentages of missed opportunities increased with the increase in the level of openness and diversity. A series of investigations helped us realize that the agents were in a long chase of chasing each other around, rather than bidding on common tasks.

After having figured out the reason behind the missed opportunities, our next contribution was to introduce and implement $p_{auctioned+}$, an enhancement to $p_{auctioned}$ Chen (2017). The goal of $p_{auctioned+}$ was to help agents choose common tasks more frequently to bid on and find the needed number of teammates, rather than chasing each other around. We subsequently conducted a series of comparison tests comparing the performances of $p_{auctioned}$ and $p_{auctioned+}$. These tests helped us realize that $p_{auctioned+}$ outperformed $p_{auctioned}$. This superior performance was because of agents realizing to bid on common tasks rather than chasing each other around. This reduction in the chase was further confirmed by analyzing the unique tasks agents were bidding for. Analyzing the unique tasks agents bid for showed that $p_{auctioned+}$ did reduce the average percentage of unique tasks agents bid for by helping them reduce the chase around.

The final contribution of this research work was to clean up the original simulation software Chen (2017). To implement the clean-up, the main modules were divided into parent and child classes and appropriate objects were made use of. Our investigations required metrics that were not computed by the original simulation software. For the computation of these new metrics we added additional modules and postprocessing scripts. This simulation software package is available for others to experiment with.

7.2 Future Work

In this Section, we chalk out our plans for the future work.

First, we studied only some AO-TO combinations ((AO, TO) = (0,0), (0.05, 0.05), and (0.1, 0.1)) instead of all the 9x9 combinations. This was done to save time as running all 9x9 combinations would have taken much more time in both simulations and subsequent analyses. We thought that by only looking at the above three combinations, we would have sufficient representation and insights. Having derived insights from the data of these combinations it would make sense to continue with all combinations in the future to fill in the gap and to further confirm the findings for various investigations.

Second, we realized that $p_{auctioned+}$ performed better than $p_{auctioned}$ as $p_{auctioned+}$ was reducing the agents chasing each other around over time. As a part of the future work, we can analyze the reason behind $p_{auctioned+}$ reducing the chase. We wish to dive deeper to investigate what was contributing to the better performance of $p_{auctioned+}$, (1) Including the information about unfilled teammate posts in $p_{auctioned+}$, or/and (2) Agents taking the hint to stay put and these actions causing the $p_{auctioned+}$ to get better.

Third, we would like to make the simulation software more robust and intuitive than it presently is. This would include incorporating abstraction, polymorphism, renaming the variable and function names, and making use of appropriate data structures.

We also plan to include an elaborate documentation for the whole codebase. This documentation would make it easier for new developers to understand and contribute to the codebase.

REFERENCES

Abdallah, S., & Lesser, V. (2004, September). Organization-based cooperative coalition formation. In Intelligent Agent Technology, 2004. (IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on (pp. 162-168). IEEE.

Albrecht, S. V., & Ramamoorthy, S. (2013, May). A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems (pp. 1155-1156). International Foundation for Autonomous Agents and Multiagent Systems.

Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Prentice-Hall series in social learning theory.

Barrett, S., Stone, P., Kraus, S., & Rosenfeld, A. (2013, July). Teamwork with Limited Knowledge of Teammates. In AAAI.

Blikstein, P., & Wilensky, U. (2005). Less is more: Agent-based simulation as a powerful learning tool in materials science. In IV International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, Holland.

Brooks, C. H., & Durfee, E. H. (2003). Congregation formation in multiagent systems. Autonomous Agents and Multi-Agent Systems, 7(1), 145-170.

Bull, S., Greer, J., McCalla, G., Kettel, L., & Bowes, J. (2001, July). User modelling in i-help: What, why, when and how. In International Conference on User Modeling (pp. 117-126). Springer Berlin Heidelberg.

Caillou, P., Aknine, S., & Pinson, S. (2002). How to Form and Restructure Multi-agent Coalitions. National Conference on Artificial Intelligence (AAAI 02) Workshop on Coalition Formation. Edmonton, Canada: AAAI Press.

Chaimontree, S., Atkinson, K., & Coenen, F. (2012). A framework for multi-agent based clustering. Autonomous Agents and Multi-Agent Systems, 25(3), 425-446.

Chalkiadakis, and G., Boutilier, C. (2008), Sequential decision making in repeated coalition formation under uncertainty. In Proceedings of the seventh international joint conference on autonomous agents and multiagent systems (AAMAS'08) (pp. 347–354).

Chalkiadakis, G. (2007). A Bayesian APPROACH TO MULTIAGENT REINFORCEMENT LEARNING AND COALITION FORMATION UNDER UNCERTAINTY (Doctoral dissertation, University of Toronto).

Chen, B., Chen, X., Timsina, A., & Soh, L. K. (2015, May). Considering agent and task openness in ad hoc team formation. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (pp. 1861-1862). International Foundation for Autonomous Agents and Multiagent Systems.

Chen, B. (2017). Investigating agent and task openness in adhoc team formation. Thesis diss., Department of Computer Science, University of Nebraska–Lincoln, 2017.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. American educator, 15(3), 6-11.

Debdi, O., Vel'zquez-Iturbide, J. Á., Esteban-S'nchez, N., & Perez-Carrasco, A. (2012, July). Observations as a method to evaluate a computer-based approach to learning algorithms. In Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on (pp. 433-435). IEEE.

Emery-Montemerlo, R., Gordon, G., Schneider, J., & Thrun, S. (2004, July). Approximate solutions for partially observable stochastic games with common payoffs. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 136-143). IEEE Computer Society.

Fernlund, H. K. G. (2004). Evolving models from observed human performance (Doctoral dissertation, University of Central Florida Orlando, Florida).

Fisch, D., Janicke, M., Kalkowski, E., & Sick, B. (2009, March). Learning by teaching versus learning by doing: Knowledge exchange in organic agent systems. In Intelligent Agents, 2009. IA'09. IEEE Symposium on (pp. 31-38). IEEE.

Floyd, M. W., & Esfandiari, B. (2011, November). A case-based reasoning framework for developing agents using learning by observation. In Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on (pp. 531-538). IEEE.

Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S., & Kettel, L. (2001). Lessons learned in deploying a multi-agent learning support system: The I-Help experience. Proceedings of AI in Education AIED, 410-421.

Griffiths, N., & Luck, M. (2003, July). Coalition formation through motivation and trust. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems (pp. 17-24). ACM.

Huynh, T. D., Jennings, N. R., & Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems, 13(2), 119-154.

Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., & Toyoda, J. I. (2000, June). How can we form effective collaborative learning groups?. In International Conference on Intelligent Tutoring Systems (pp. 282-291). Springer Berlin Heidelberg.

Isaac, A., & Sammut, C. (2003). Goal-directed learning to fly. In ICML (pp. 258-265).

Jamroga, W., Męski, A., & Szreter, M. (2013). Modularity and openness in modeling multi-agent systems. arXiv preprint arXiv:1307.4477.

Jia-hai, Y., Shun-kun, Y., & Zhao-guang, H. (2005, November). A multi-agent trading platform for electricity contract market. In Power Engineering Conference, 2005. IPEC 2005. The 7th International (pp. 1024-1029). IEEE.

JIANG, J. G., Zhao-Pin, S. U., Mei-Bin, Q. I., & ZHANG, G. F. (2008). Multi-task coalition parallel formation strategy based on reinforcement learning. Acta Automatica Sinica, 34(3), 349-352.

Johnson, C. L., & Gonzalez, A. J. (2014). Learning collaborative team behavior from observation. Expert Systems with Applications, 41(5), 2316-2328.

Jumadinova, J., Dasgupta, P., & Soh, L. K. (2014). Strategic capability-learning for improved multiagent collaboration in ad hoc environments. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44(8), 1003-1014.

Kenari, S. M. S., Jahan, M. V., & Jalali, M. (2011, June). Multi-skill agents coalition formation under skill uncertainty. In Artificial Intelligence and Signal Processing (AISP), 2011 International Symposium on (pp. 89-96). IEEE.

Khandaker, N., & Soh, L. K. (2010, May). ClassroomWiki: a wiki for the classroom with multiagent tracking, modeling, and group formation. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1 (pp. 1377-1378). International Foundation for Autonomous Agents and Multiagent Systems.

Klusch, M., & Gerber, A. (2002). Dynamic coalition formation among rational agents. IEEE Intelligent Systems, 17(3), 42-47.

Kraus, S., Shehory, O., & Taase, G. (2003, July). Coalition formation with uncertain heterogeneous information. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems (pp. 1-8). ACM.

Kraus, S., Shehory, O., & Taase, G. (2004, July). The advantages of compromising in coalition formation with incomplete information. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2 (pp. 588-595). IEEE Computer Society.

Lee, K., Su, Y., Kim, T. K., & Demiris, Y. (2013). A syntactic approach to robot imitation learning using probabilistic activity grammars. Robotics and Autonomous Systems, 61(12), 1323-1334.

Liemhetcharat, S., & Veloso, M. (2012, June). Modeling and learning synergy for team formation with heterogeneous agents. In Proceedings of the 11th International

Conference on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 365-374). International Foundation for Autonomous Agents and Multiagent Systems.

Maret, P., Hammond, M., & Calmet, J. (2004). Agent Societies for Corporate Knowledge Issues. In The Fifth International Workshop on Engineering Societies in the Agents World (ESAW).

Merida-Campos, C., & Willmott, S. (2004, July). Modelling coalition formation over time for iterative coalition games. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2 (pp. 572-579). IEEE Computer Society.

Moura, I., & Sarma, V. V. S. (2005). Multiagent System and Imitative Consciousness. In IICAI (pp. 2956-2975).

O. Shehory and S. Kraus (1998), Methods for task allocation via agent coalition formation, Artificial Intelligence 101 (1–2) (pp. 165–200).

Pinyol, I., & Sabater-Mir, J. (2013). Computational trust and reputation models for open multi-agent systems: a review. Artificial Intelligence Review, 40(1), 1-25.

Procaccia, A. D., & Rosenschein, J. S. (2006, May). The communication complexity of coalition formation among autonomous agents. In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (pp. 505-512). ACM

Ramchurn, S. D., Polukarov, M., Farinelli, A., Truong, C., & Jennings, N. R. (2010, May). Coalition formation with spatial and temporal constraints. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 3-Volume 3 (pp. 1181-1188). International Foundation for Autonomous Agents and Multiagent Systems.

Ray, D., & Vohra, R. (2014). Coalition formation. Handbook of Game Theory, 4, 239-326.

Roediger III, H. L. and M. A. Smith (2012). The "pure-study" learning curve: The learning curve without cumula-tive testing. Memory and Cognition, 40:989-1002.

S. Liemhetcharat and M. Veloso (2011), Modeling mutual capabilities in heterogeneous teams for role assignment, Proc. IEEE/RSJ International Conference Intelligent Robots Syst., pp. 3638-3644.

Sammut, C., Hurst, S., Kedzier, D., & Michie, D. (2014, June). Learning to fly. In Proceedings of the ninth international workshop on Machine learning (pp. 385-393).

Schatten, M. (2014). Smart residential buildings as learning agent organizations in the internet of things. Business Systems Research Journal, 5(1), 34-46.

Schön, D. A. (1987). Educating the reflective practitioner: Toward a new design for teaching and learning in the professions. Jossey-Bass.

Shehory, O. (2001). Software architecture attributes of multi-agent systems. In Agent-Oriented Software Engineering (pp. 77-90). Springer Berlin Heidelberg.

Shehory, O., Sycara, K., & Jha, S. (1998b). Multi-agent coordination through coalition formation. Intelligent agents IV agent theories, architectures, and languages, 143-154.

Shintani, T., & Ito, T. (1998). An architecture for multi-agent negotiation using private preferences in a meeting scheduler. PRICAI'98: Topics in Artificial Intelligence, 47-58.

Soriano Marcolino, L., Xin Jiang, A., & Tambe, M. (2013). Multi-agent team formation: diversity beats strength?. IJCAI.

Steinhage, A., & Bergener, T. (2000). Learning by doing: A dynamic architecture for generating adaptive behavioral sequences. In Proceedings of the Second International ICSC Symposium on Neural Computation (NC) (pp. 813-820).

Stone, P., & Kraus, S. (2010, May). To teach or not to teach?: decision making under uncertainty in ad hoc teams. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1 (pp. 117-124). International Foundation for Autonomous Agents and Multiagent Systems.

Sun, R. (2001). Cognitive science meets multi-agent systems: A prolegomenon. Philosophical psychology, 14(1), 5-28.

Van De Vijsel, M., & Anderson, J. (2004). Coalition formation in multi-agent systems under real-world conditions. Proceedings of association for the advancement of artificial intelligence.

Vassileva, J., Breban, S., & Horsch, M. (2002). Agent Reasoning Mechanism for Long–Term Coalitions Based on Decision Making and Trust. Computational Intelligence, 18(4), 583-595.

Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C., & Grant, S. (1999, July). A multi-agent approach to the design of peer-help environments. In Proceedings of AIED (Vol. 99, pp. 38-45).

Vassileva, J., McCalla, G., & Greer, J. (2003). Multi-agent multi-user modeling in I-Help. User Modeling and User-Adapted Interaction, 13(1), 179-210.

Wifall, T., B. McMurray, and E. Hazeltine (2014). Percep-tual similarity affects the learning curve (but not necessari-ly learning), Journal of Experimental Psychology: Gen-eral, 143:312-331.

Wooldridge, M. (2009). An Introduction to Multiagent Systems. Chichester, U.K: Wiley.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2), 115–152. https://doi.org/10.1017/S0269888900008122

Wu, F., Zilberstein, S., & Chen, X. (2011, July). Online planning for ad hoc autonomous agent teams. In IJCAI (pp. 439-445).

Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., & Iwasaki, A. (2005, July). Coalitional games in open anonymous environments. In AAAI (Vol. 5, pp. 509-514).

APPENDIX A

How $p_{auctioned+}$ performs better than $p_{auctioned}$

In this Appendix, we investigate how the new solution $p_{auctioned+}$ as described in Chapter Methodology Equation 3.10 performs better than $p_{auctioned}$, as seen in Chapter 5. Recall that $p_{auctioned+}$ is design to help maximizing utility even in an agent- and task-diverse environment, specifically at reducing or preventing agents from "chasing each other around" while submitting bids to successfully complete tasks.

A.1 Comparisons between $p_{auctioned+}$ and $p_{auctioned}$

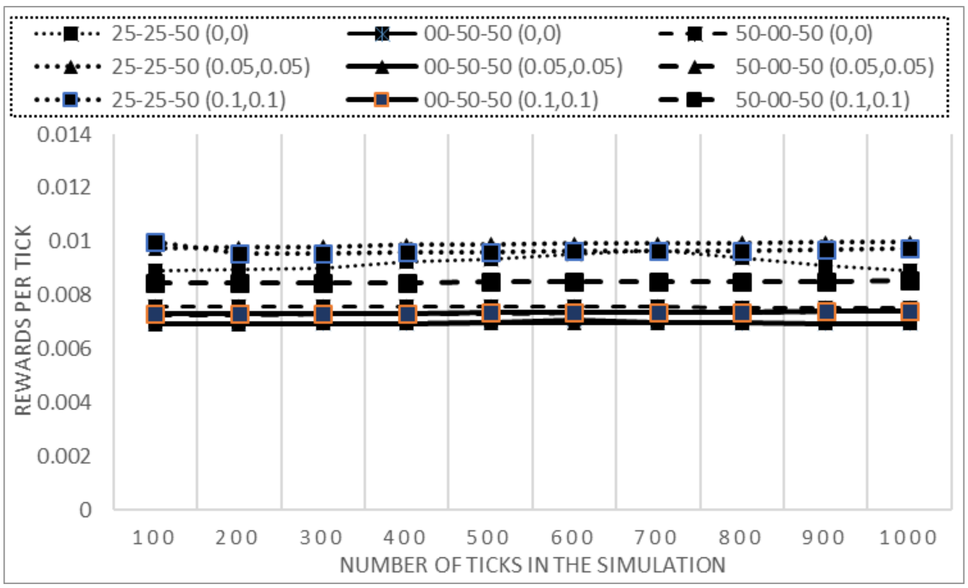
In this section, we carry out a series of investigations to study how $p_{auctioned+}$ is better than $p_{auctioned}$. We conducted tests to analyze the average rewards per tick, average rewards per task, average percentage of tasks auctioned off, average learning gains per tick, and average learning gains per task. The motive of introducing $p_{auctioned+}$ was to help make the environment robust and tackle diversity, and in return get more tasks auctioned off timely.

A.1.1 Impacts on the Rewards earned

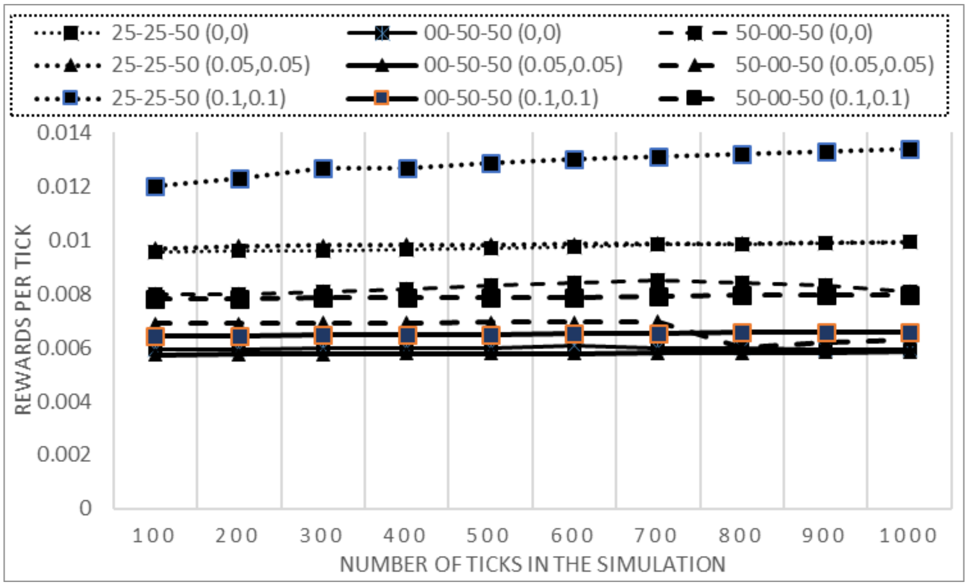
In this Section, we analyze the impacts of both $p_{auctioned+}$ and $p_{auctioned}$ on the rewards earned by the agents per tick and task respectively.

A.1.1.1 Rewards earned per tick

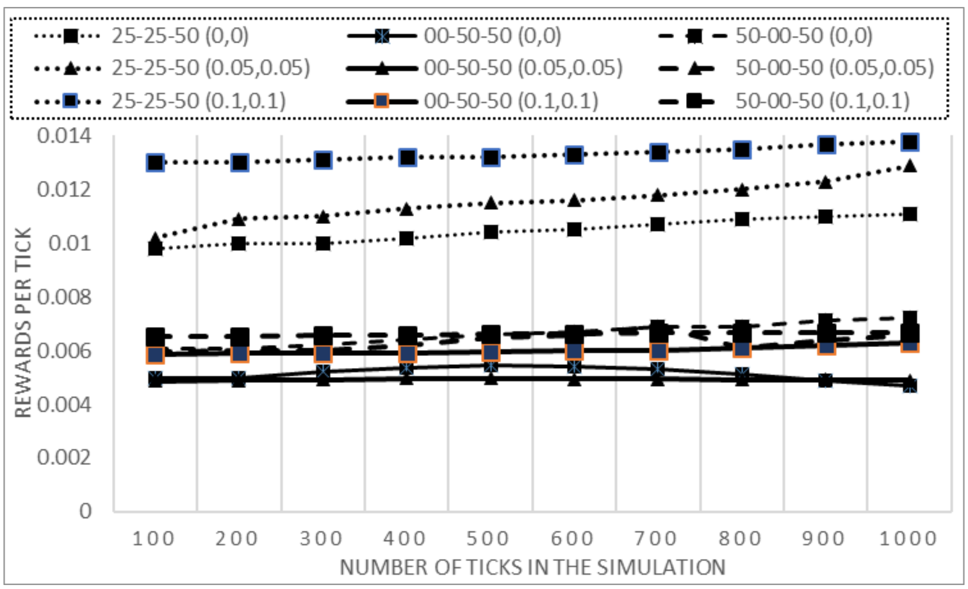
In this Section, we analyze the impacts both $p_{auctioned+}$ and $p_{auctioned}$ have on the average rewards earned per tick. Figure A.1, A.2, A.4, and A.5 present the average rewards earned per tick and per task for AO = TO = 0, 0.05, and 0.1, TD = Low, Medium, and High, and AD = 25-25-50, 00-50-50, and 50-00-50, respectively for $p_{auctioned+}$ and $p_{auctioned}$.



(a) TD = Low

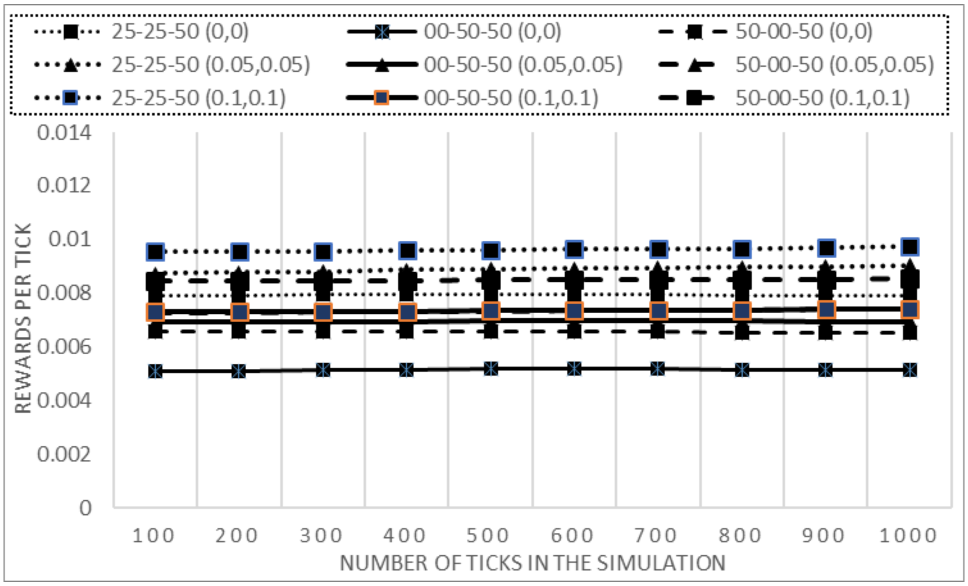


(b) TD = Medium

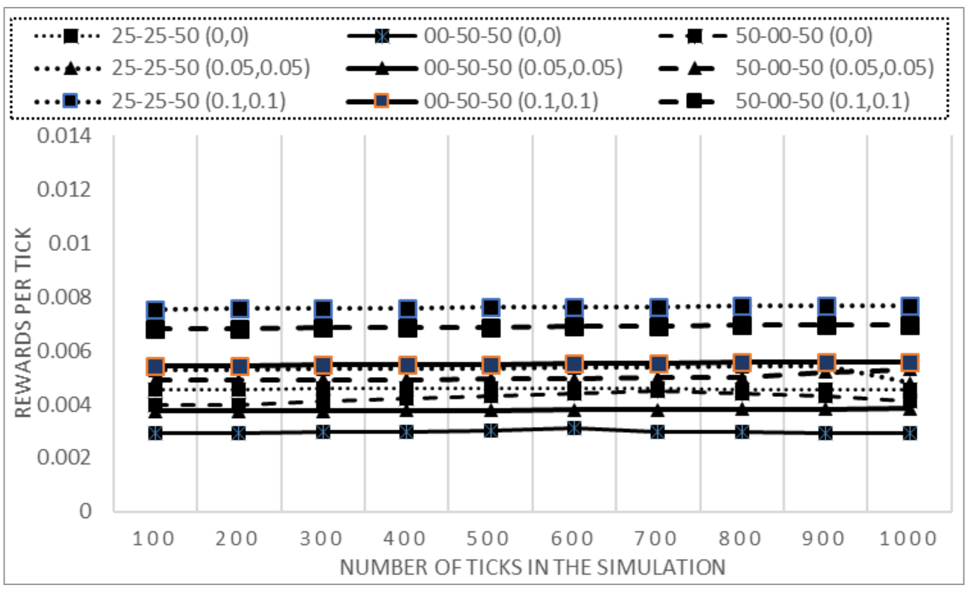


(c) TD = High

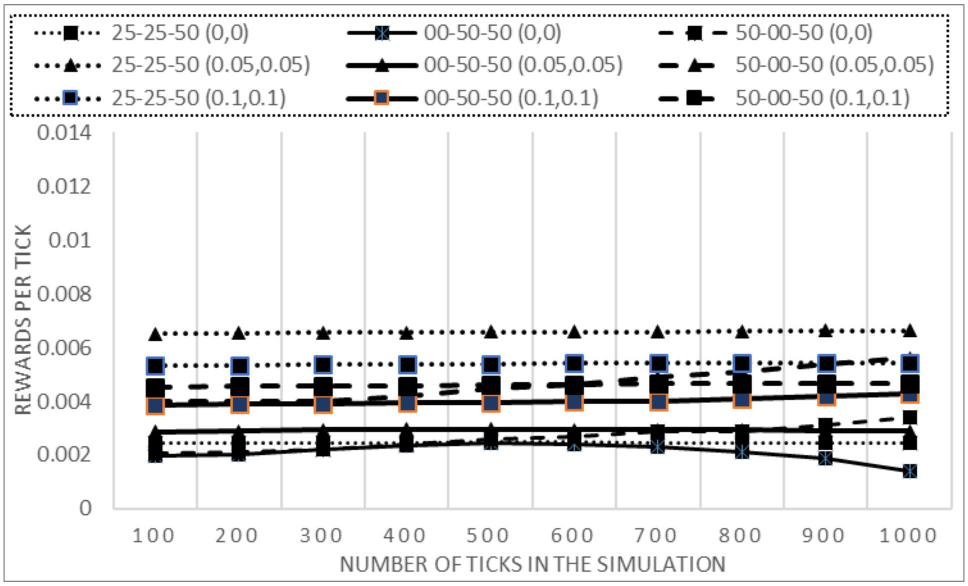
Figure A.1. Average rewards earned per tick, AO = TO = 0, 0.05, and 0.1, TD = Low, Medium, and High, AD = 25-25-50, 00-50-50, and 50-00-50 for $p_{auctioned}$, $p < 0.001$.



(a) TD = Low.



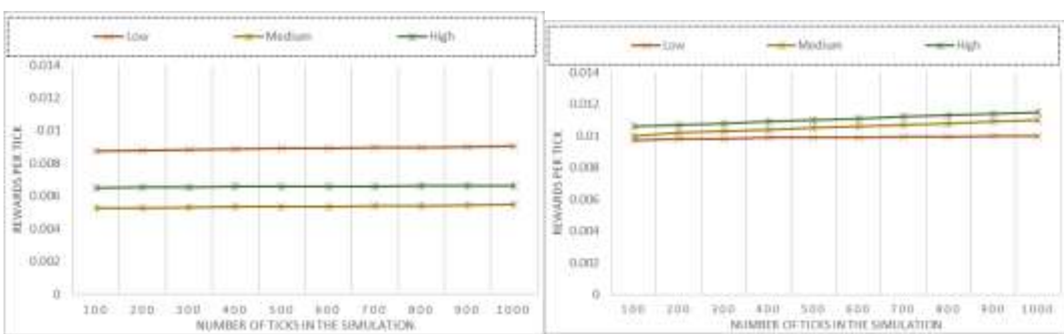
(b) TD = Medium.



(c) TD = High.

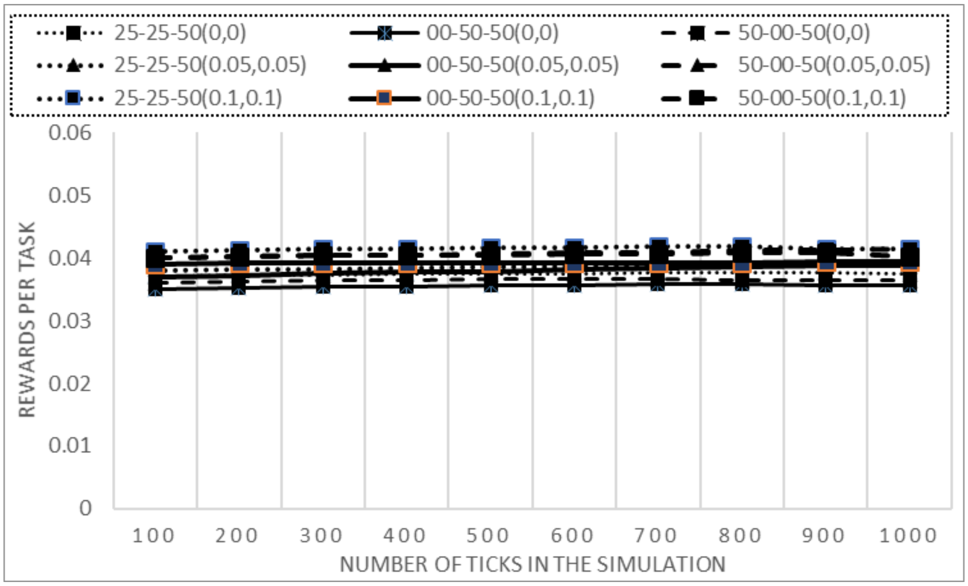
Figure A.2. Average rewards earned per tick, AO = TO = 0, 0.05, and 0.1, TD = Low, Medium, and High, AD = 25-25-50, 00-50-50, and 50-00-50 for $p_{auctioned+}$, $p < 0.001$.

In order to be able to analyze Figures A.1 and A.2 in detail, we prune out AD = 25-25-50, AO = TO = 0.05, and TD = Low, Medium, and High and present the same results in Figure A.3.

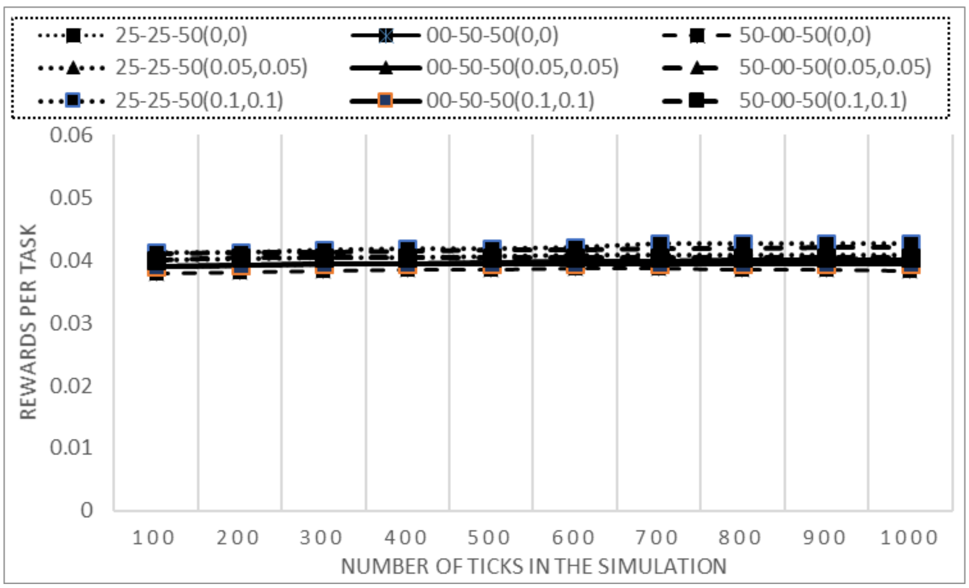


(a) $p_{auctioned}$ (b) $p_{auctioned+}$

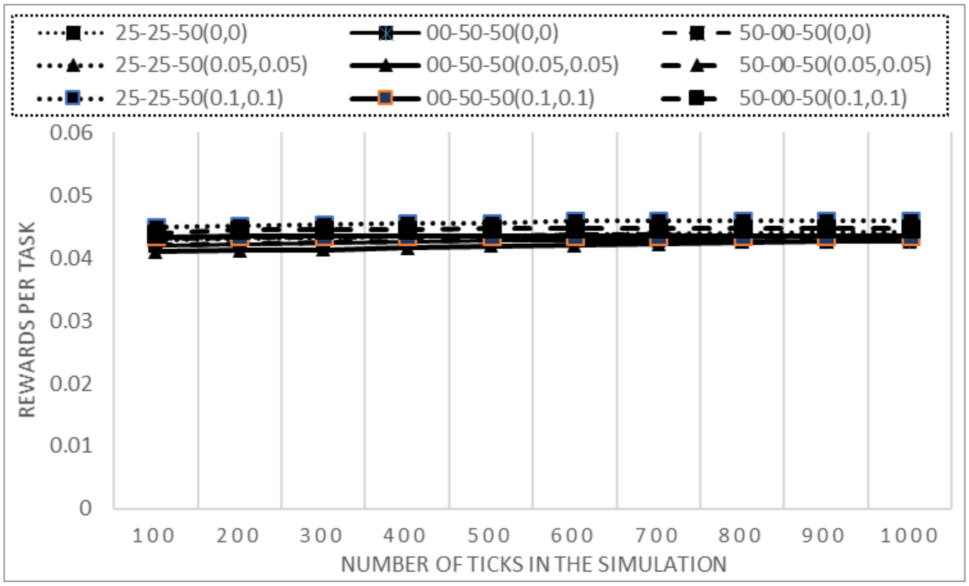
Figure A.3. Average rewards earned per tick, AO = TO = 0.05, TD = Low, Medium, and High, AD = 25-25-50.



(a) TD = Low

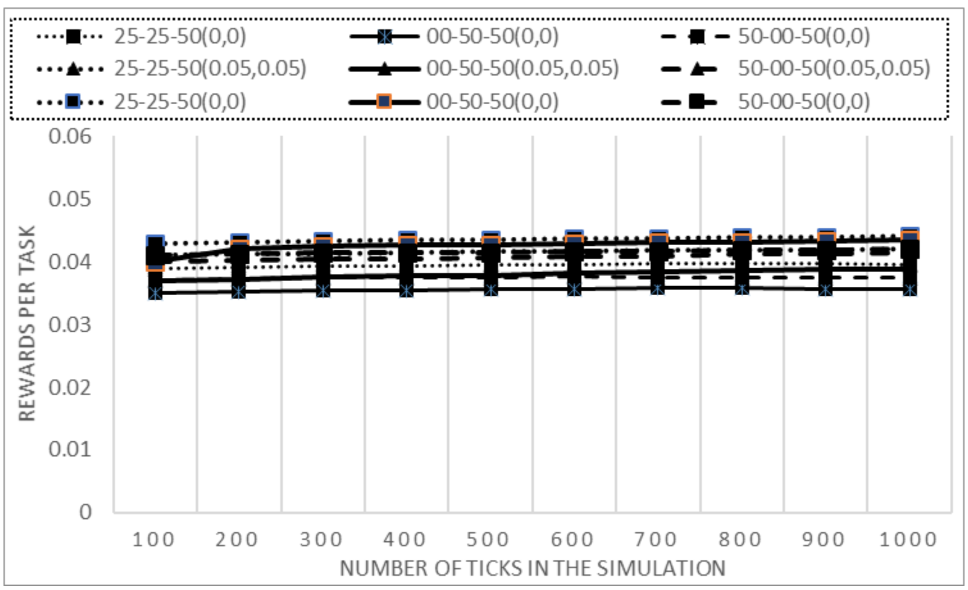


(b) TD = Medium

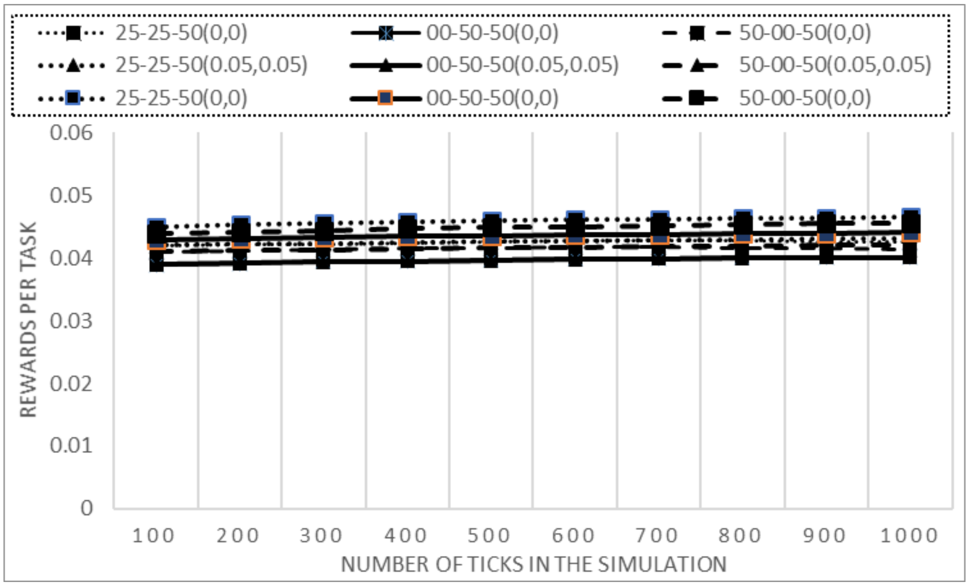


(c) TD = High

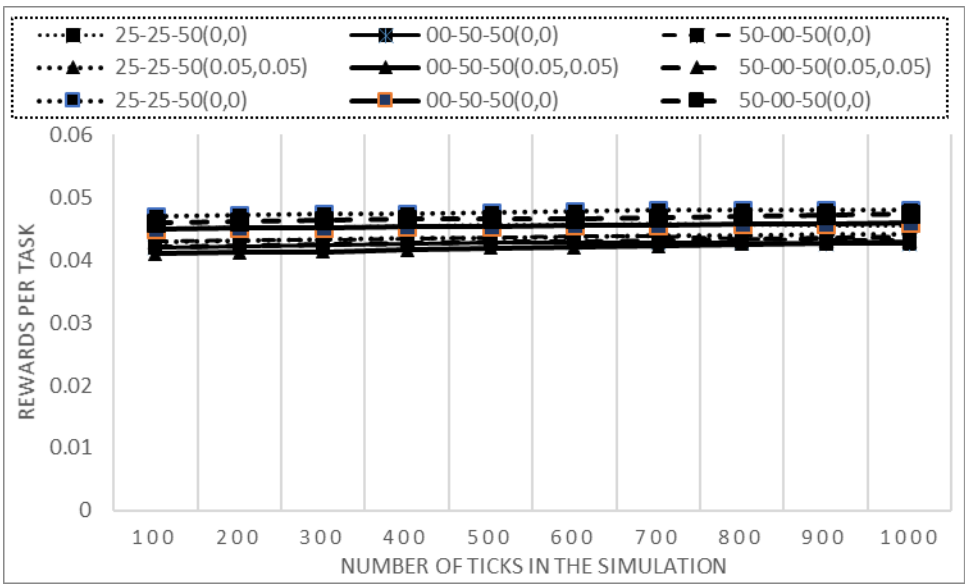
Figure A.4. Average rewards earned per task, AO = TO = 0, 0.05, and 0.1, TD = Low, Medium, and High, AD = 25-25-50, 00-50-50, and 50-00-50, $p_{auctioned}$, $p < 0.001$.



(a) TD = Low



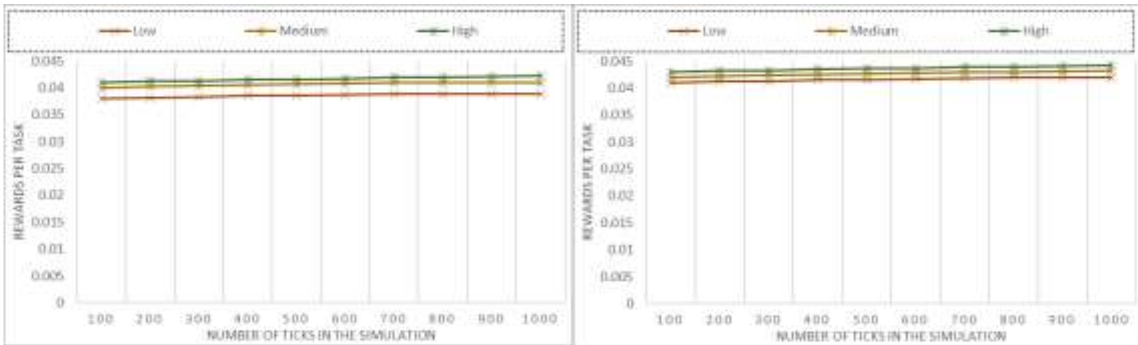
(b) TD = Medium



(c) TD = High

Figure A.5. Average rewards earned per task, AO = TO = 0, 0.05, and 0.1, TD = Low, Medium, and High, AD = 25-25-50, 00-50-50, and 50-00-50, $p_{auctioned+}$, $p < 0.001$.

In order to be able to analyze Figures A.4 and A.5 in detail, we prune out AD = 25-25-50, AO = TO = 0.05, and TD = Low, Medium, and High and present the same results in Figure A.6.



(a) $p_{auctioned}$ (b) $p_{auctioned+}$

Figure A.6. Average rewards earned per task, AO = TO = 0.05, TD = Low, Medium, and High, AD = 25-25-50, $p < 0.001$.

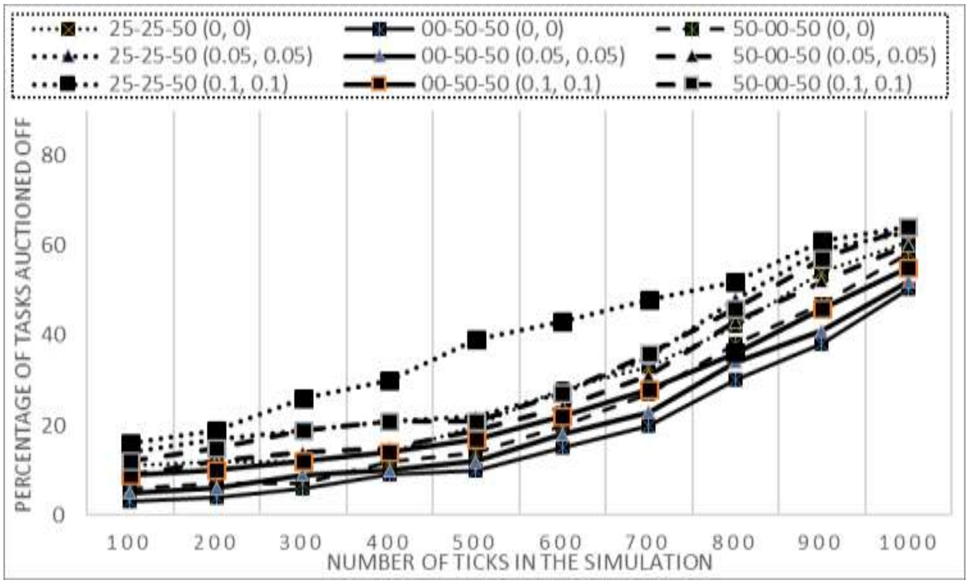
As seen from Figures A.1, A.2, A.4, and A.5, $p_{auctioned+}$ leads to better average rewards earned per tick and task as compared to $p_{auctioned}$. This is further confirmed from Figures A.3 and A.6. It can also be seen that for both the methods the trends on the rewards earned is the same for diversity and openness.

Diversity. The average rewards are the highest for the 25-25-50 configuration, followed by the 50-00-50, and then the 00-50-50 configuration. As seen earlier, a highly diverse configuration helps more tasks get auctioned off and since the learning gaps between Apprentices and Generalists is more ideal as compared to Apprentices and Specialists, hence the 50-00-50 does better than 00-50-50. As discussed in Chapter Methodology, we referred to Bandura’s theory for calculating learning by observation. Based on Equation (3.3), we see that if the difference between an observer’s expertise level and a performer’s is too large or too small, then the observer does not benefit much from learning by observation. Thus, it is easier for an Apprentice to learn from a Generalist as compared to a Specialist, and a Generalist to learn from a Specialist. Therefore, the rewards are higher for the 50-00-50 than the 00-50-50 configuration. The average rewards increased with the level of task diversity as the agents could get a wide variety of tasks auctioned off because they were able to learn a wide variety of capabilities.

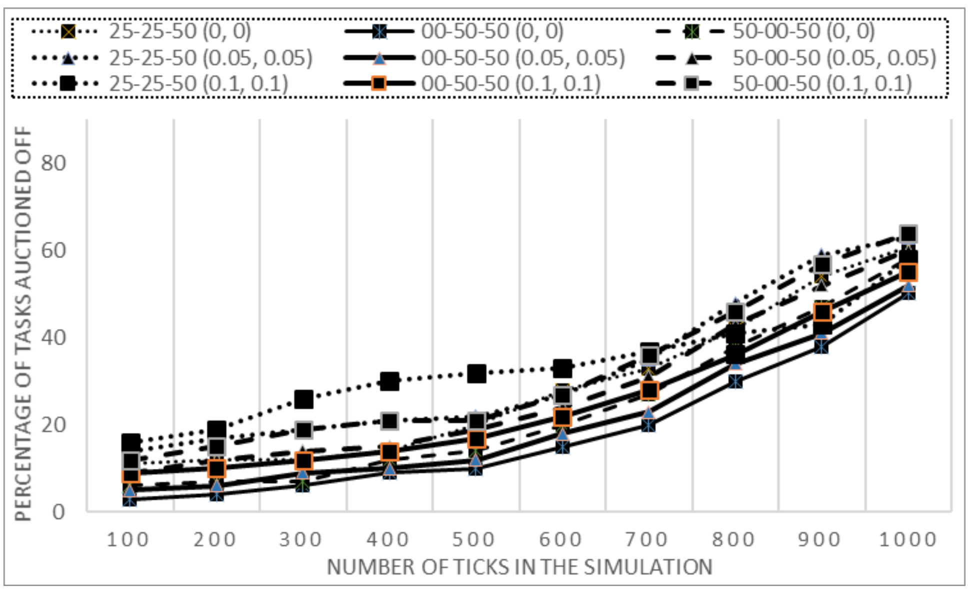
Openness. As the openness increases the average rewards earned increase as well. Task diversity helps agents find tasks suitable for them as opposed to when there is no task diversity and if some agent doesn't find a suitable task then the chances of the agent finding a task throughout the auction rounds reduces. Agent diversity helps agents of different expertise to keep coming together and helping to get tasks auctioned off.

A.1.2 Impacts on Percentage of Tasks Auctioned off

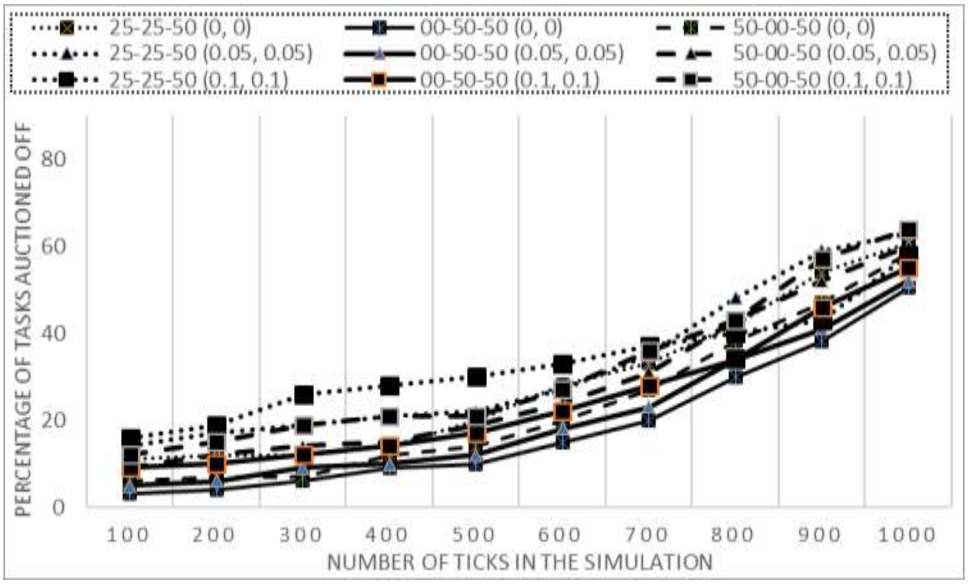
Here we analyze the impacts on the average percentage of tasks auctioned off. To do so, Figure A.7 presents the average percentage of tasks auctioned off for TD = Low, Medium, High with AD = 25-25-50, 00-50-50, and 50-00-50, and AO = TO = 0, 0.05, and 0.1, respectively for $p_{auctioned}$ and $p_{auctioned+}$.



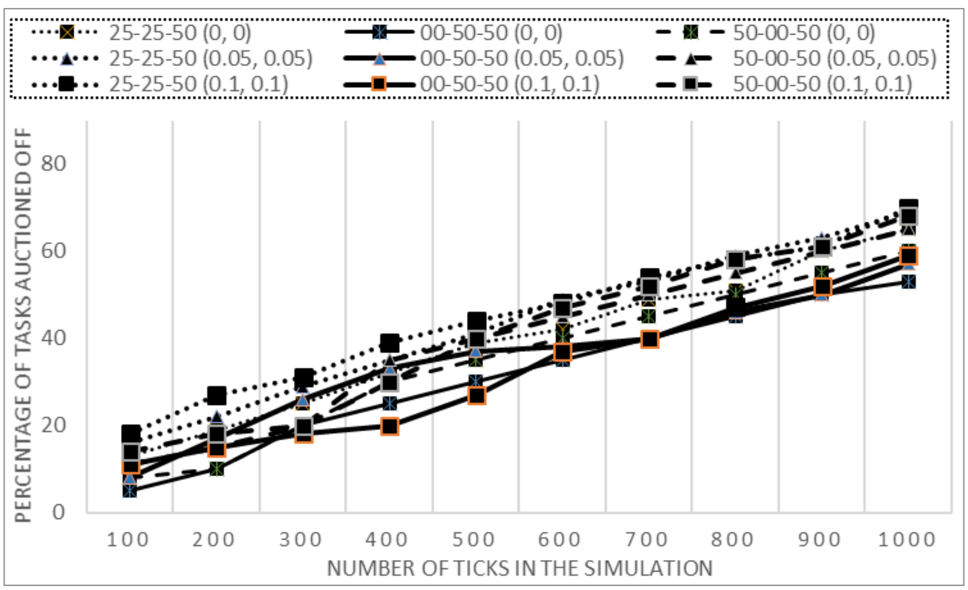
(a) TD = Low ($p_{auctioned}$)



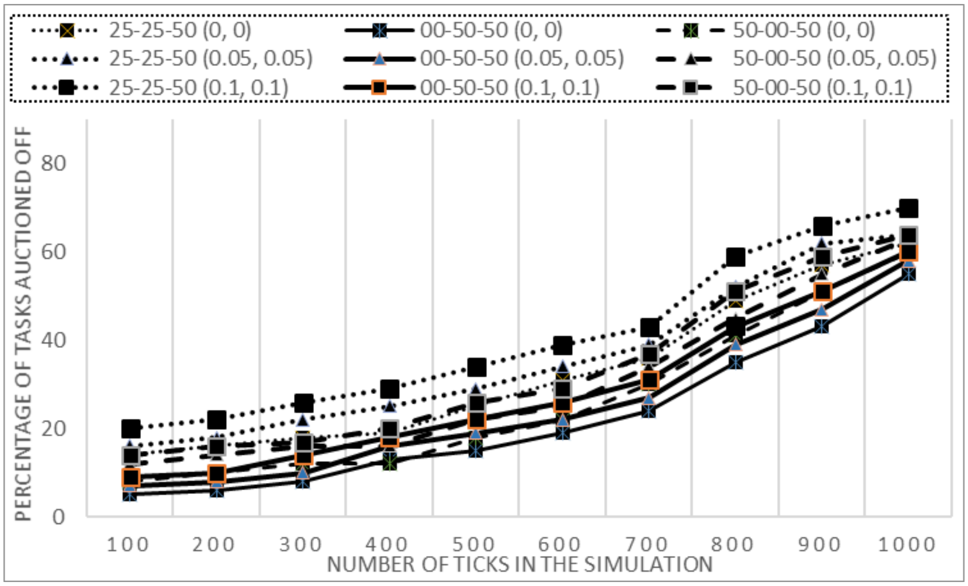
(b) TD = Medium ($p_{auctioned}$)



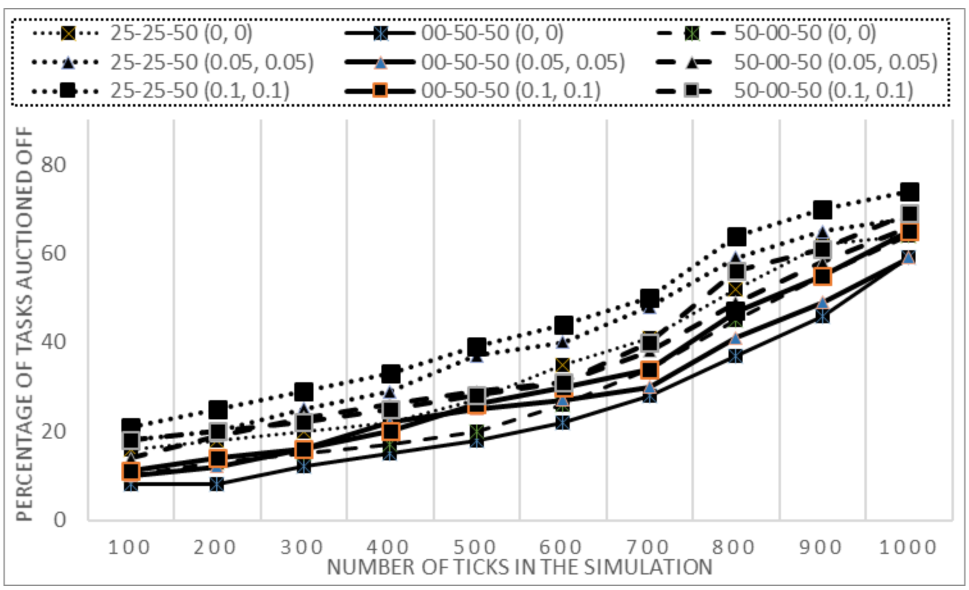
(c) TD = High ($p_{auctioned}$)



(d) TD = Low ($p_{auctioned+}$)



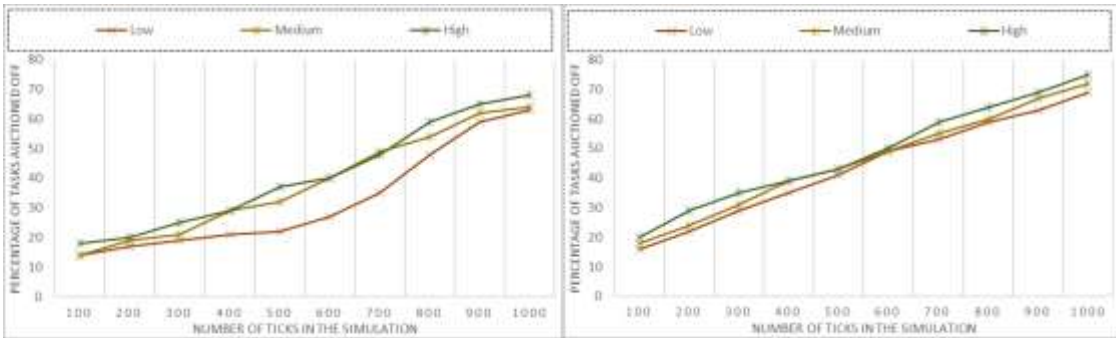
(e) TD = Medium ($p_{auctioned+}$)



(f) TD = High ($p_{auctioned+}$)

Figure A.7. Average percentages of tasks auctioned off for TD = Low, Medium, and High, for AD = 25-25-50, 00-50-50, and 50-00-50, with AO = TO = 0, 0.05, and 0.1, respectively, for $p_{auctioned}$, $p_{auctioned+}$, $p < 0.001$.

To be able to better observe the results plotted in Figure A.7, we focus on TD = Low, Medium, and High, with AO = TO = 0.05, and AD = 25-25-50 in Figure A.8 to better understand the graphs.



(a) $p_{auctioned}$ (b) $p_{auctioned+}$

Figure A.8. Average percentages of Tasks auctioned off for TD = Low, Medium, and High, with AO = TO = 0.05, and AD = 25-25-50, $p < 0.001$.

As seen from Figure A.7, we realize that $p_{auctioned+}$ does better than $p_{auctioned}$. This is further proved by Figure A.8 as well. We notice that $p_{auctioned+}$ gets more tasks auctioned off than does $p_{auctioned}$. This is because with $p_{auctioned+}$ the agents were realizing to stop chasing each other and to be focusing on tasks which will see the right number of teammates needed for the task to be auctioned off. Thus, resulting in increasing the average percentage of tasks auctioned off. We also notice the same diversity and openness impact trends on both the methods:

Diversity. The average percentage of tasks auctioned off is the highest for the 25-25-50 configuration, followed by the 50-00-50 and then the 00-50-50 configuration. This happens because the 25-25-50 is the most diverse configuration and hence the agents get to work with a diverse agent set and in return be able to learn a wide variety of capabilities and different levels of expertise. 50-00-50 does better than 00-50-50 because the Generalists can bid and win a wider variety of tasks and hence help the Apprentices get better at a wider variety. As opposed to the Specialists who are good at fewer capabilities as compared to the Generalists. With the decrease in the level of task diversity, the average percentage of tasks auctioned off decreased as well. This is because in a low level of task diversity, if agents do not find tasks suitable for them then it is likely that they will find it difficult to find tasks throughout the auction rounds.

Openness. The AO = TO = 0.1 configurations gets the most tasks auctioned off followed by the 0.05 and then the configuration with no openness. A more open environment sees a flow of different capabilities at different level of expertise into the environment which helps the agents learn a wider variety of capabilities at different level of expertise. As compared to when the environment does not have openness since there are no new agents joining or old agents leaving the environment. Thus, the agents keep working with the same set of capabilities and hence do not sharpen a wider set of capabilities as compared to an open environment.

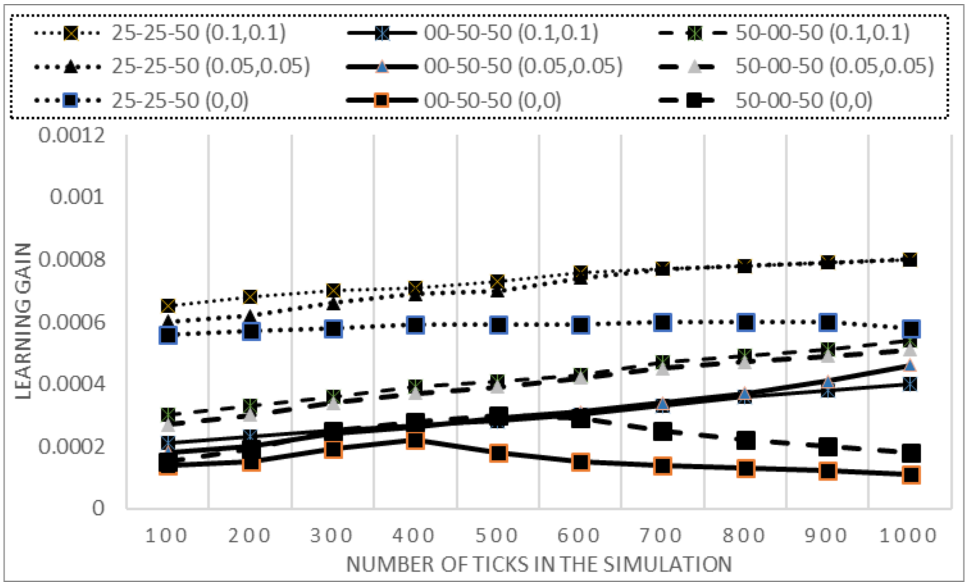
A.1.3 Impacts on the Learning Gains

This Sections helps us analyze the impacts of both $p_{auctioned+}$ and $p_{auctioned}$ on the learning gains.

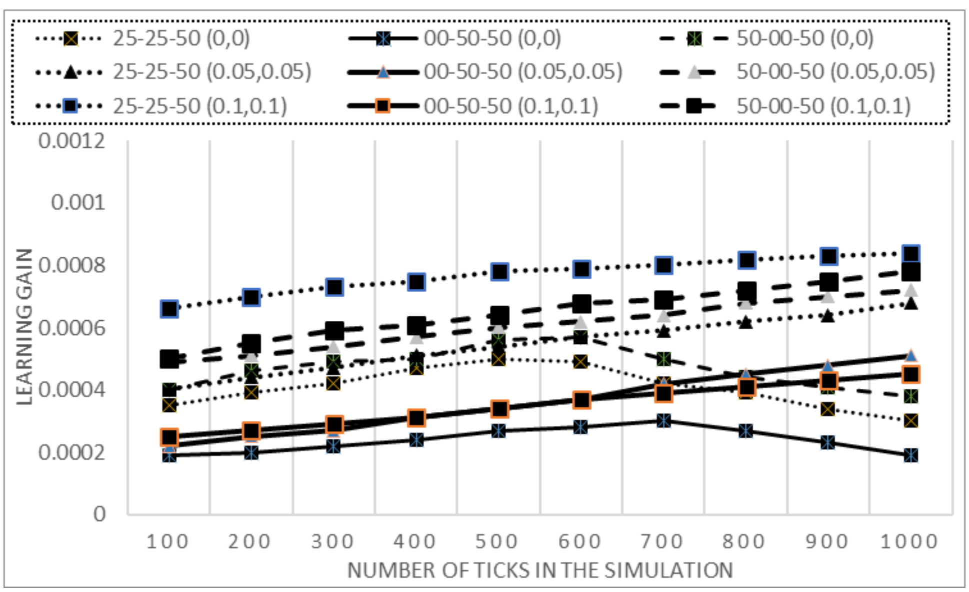
A.1.3.1 Learning Gains per Tick

Here we analyze the impacts of diversity and openness with $p_{auctioned+}$ on the learning gains. Figure A.9 presents the learning gains for TD = Low, Medium, and High, with AO = TO = 0, 0.05, and 0.1 and AD = 25-25-50, 00-50-50, and 50-00-50, respectively with $p_{auctioned+}$. Figure A.10 presents the learning gains for TD = Low, Medium, and High, with AO = TO = 0, 0.05, and 0.1 and AD = 25-25-50, 00-50-50, and 50-00-50, respectively with $p_{auctioned}$. To be able to analyze these graphs in more detail we pick AD

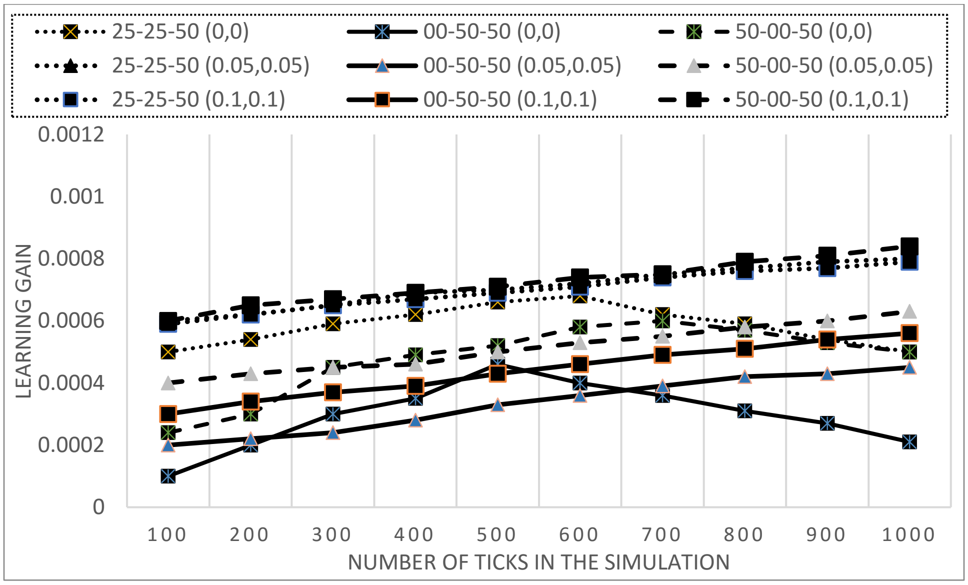
= 25-25-50 configuration for TD = Low, Medium, and High, with AO = TO = 0.05 and plot the same results in Figure A.11.



(a) TD = Low

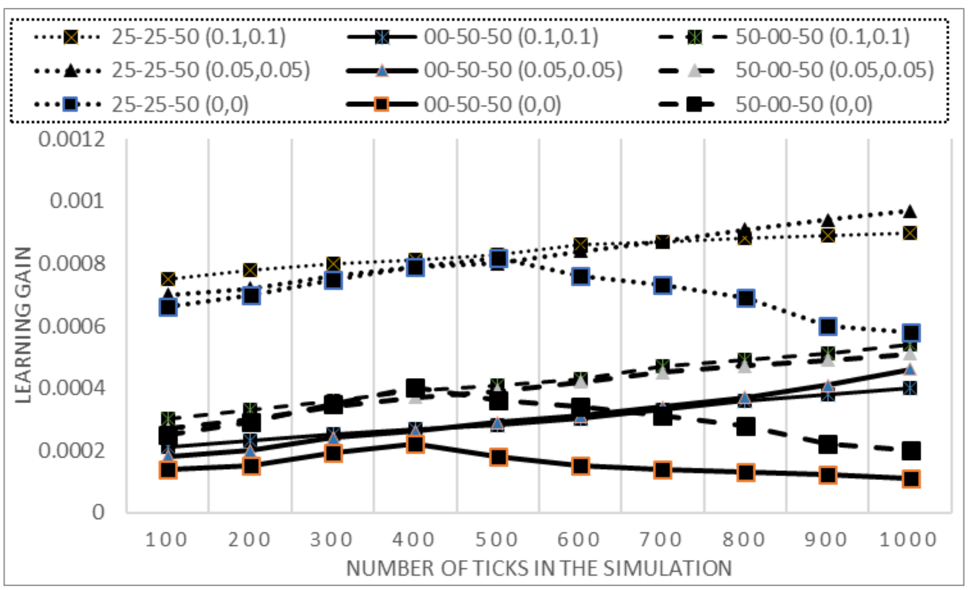


(b) TD = Medium

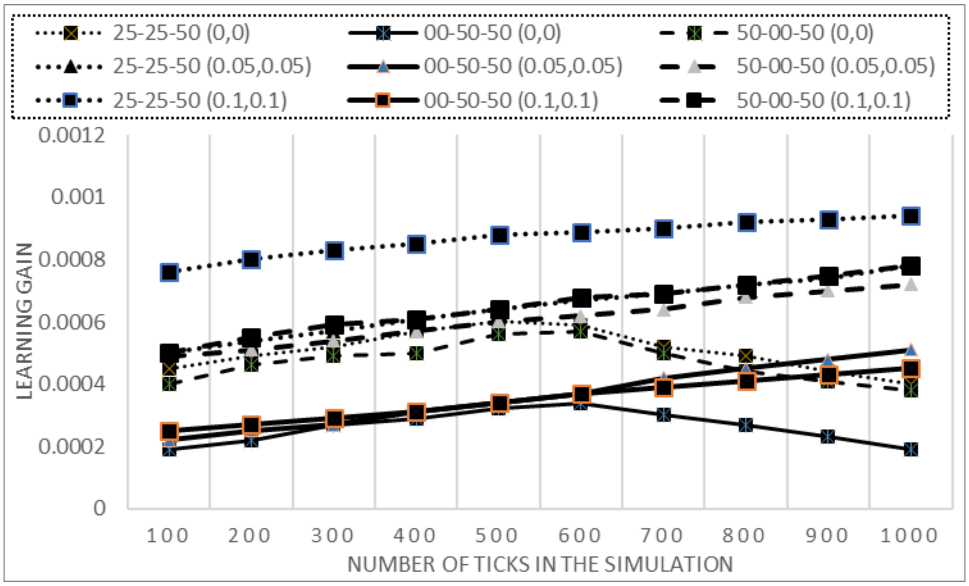


(c) TD = High

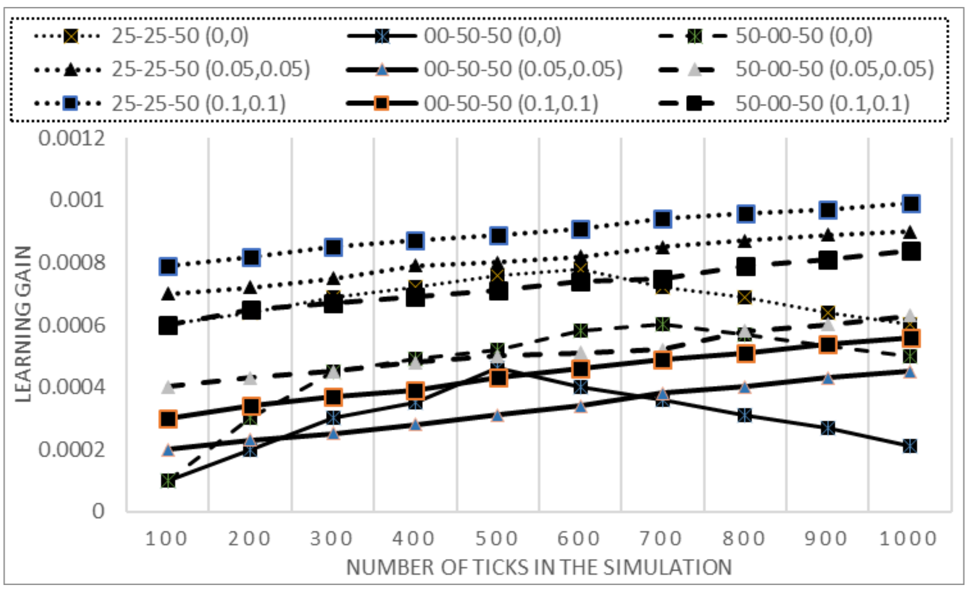
Figure A.9. Average Learning gains per tick for TD = Low, Medium, and High, for AD = 25-25-50, 00-50-50, and 50-00-50, with AO = TO = 0, 0.05, and 0.1, respectively, for $p_{auctioned}$, $p < 0.001$.



(a) TD = Low

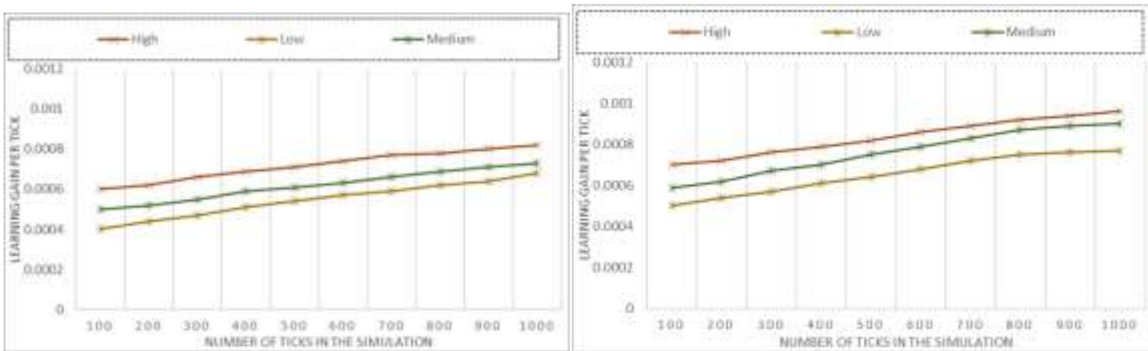


(b) TD = Medium



(c) TD = High

Figure A.10. Average Learning gains per tick for TD = Low, Medium, and High, for AD = 25-25-50, 00-50-50, and 50-00-50, with AO = TO = 0, 0.05, and 0.1, respectively, for $p_{auctioned+}$, $p < 0.001$.



(a) $p_{auctioned}$ (b) $p_{auctioned+}$

Figure A.11. Average Learning gains per tick for AO = TO = 0.05, TD = Low, Medium, and High, AD = 25-25-50, $p < 0.001$.

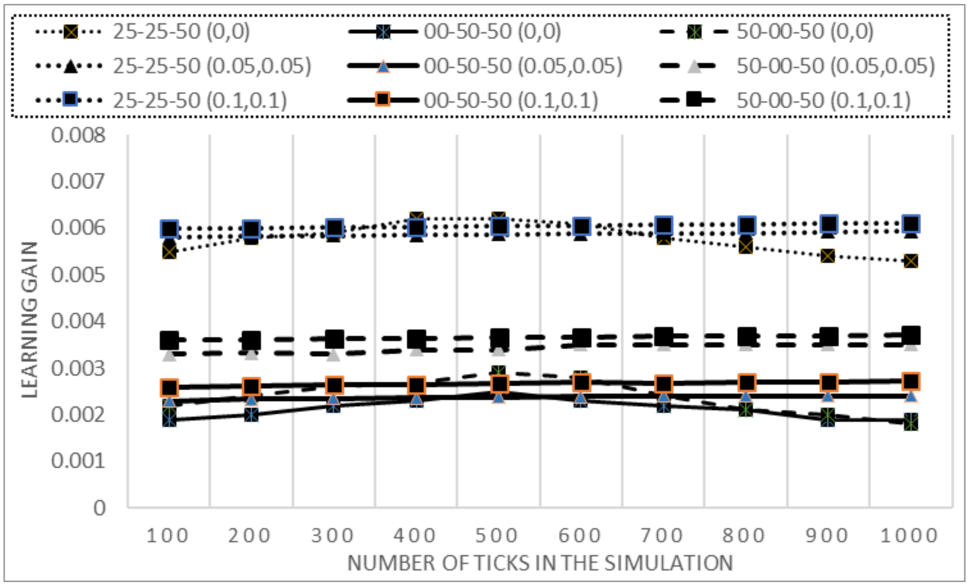
From Figures A.9 and A.10 we notice that the learning gains are better with $p_{auctioned+}$ as compared to $p_{auctioned}$. This is also confirmed from Figure A.11 which shows that the learning gains per tick is higher for $p_{auctioned+}$ than for $p_{auctioned}$. We see this because with $p_{auctioned+}$ we have more tasks that get auctioned off and also presents the agents with a chance to learn by observing and doing. Both methods show the same trends in presence of openness and diversity:

Diversity. It can be seen that the agents learn the most in the most diverse environment, which is the 25-25-50 configuration, this is because in a more diverse environment there is more diverse set of capabilities available, which helps the agents learn a diverse set of capabilities. Higher levels of task diversity also helped agents learn more, since a wider variety of tasks helped agents get better at a wider variety of capabilities. Thus, a more diverse environment helps the agent nurture through learning from a wider variety of teammates and in return get more tasks auctioned off. It can also be observed that with the increase in the level of task diversity, the average learning gains increased as well. This is because, a high task diversity presents the agents with an opportunity to be able to learn a wide variety of capabilities and in return increase their learning gains.

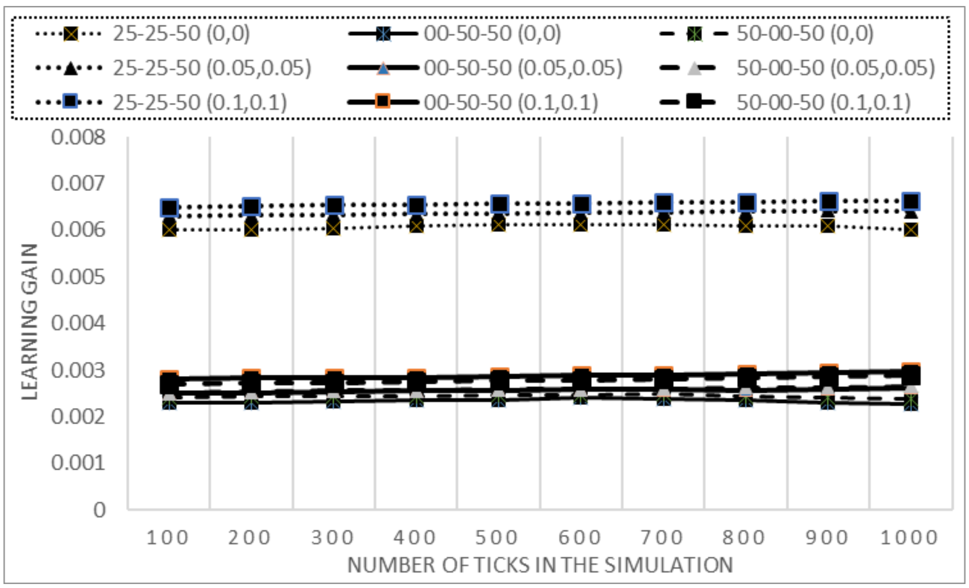
Openness. The more open an environment, the more agents learn. Thus, as seen the AO = TO = 0.1, saw the highest learning gain followed by the 0.05 configuration followed by no openness. This is because in an open environment, agents do not keep working with the same set of agents or on the same set of tasks. Openness presents them with an opportunity to work with a varied set of teammates on different sets of tasks which helps them get better at a wider variety of capabilities. Thus, in an open environment, agents can learn more from a wider variety of teammates and different tasks.

A.1.3.2 Learning Gains per Task

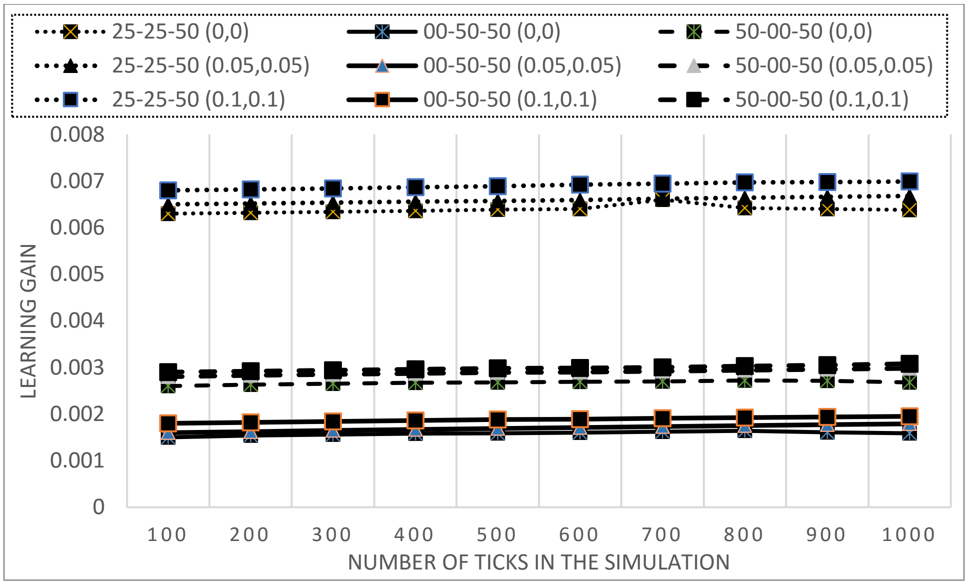
Figure A.12 presents the learning gains per task for all three TD configurations Low, Medium, and High, with AO = TO = 0, 0.05, and 0.1, and AD = 25-25-50, 00-50-50, and 50-00-50, respectively, with $p_{auctioned+}$. Figure A.13 presents the learning gains per task for all three TD configurations Low, Medium, and High, with AO = TO = 0, 0.05, and 0.1, and AD = 25-25-50, 00-50-50, and 50-00-50, respectively, with $p_{auctioned}$.



(a) TD = Low

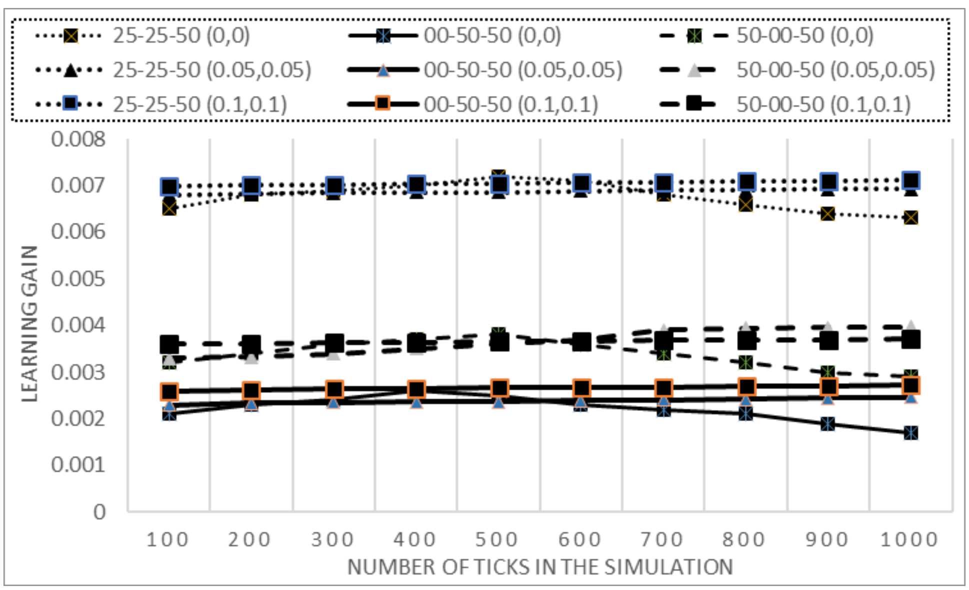


(b) TD = Medium

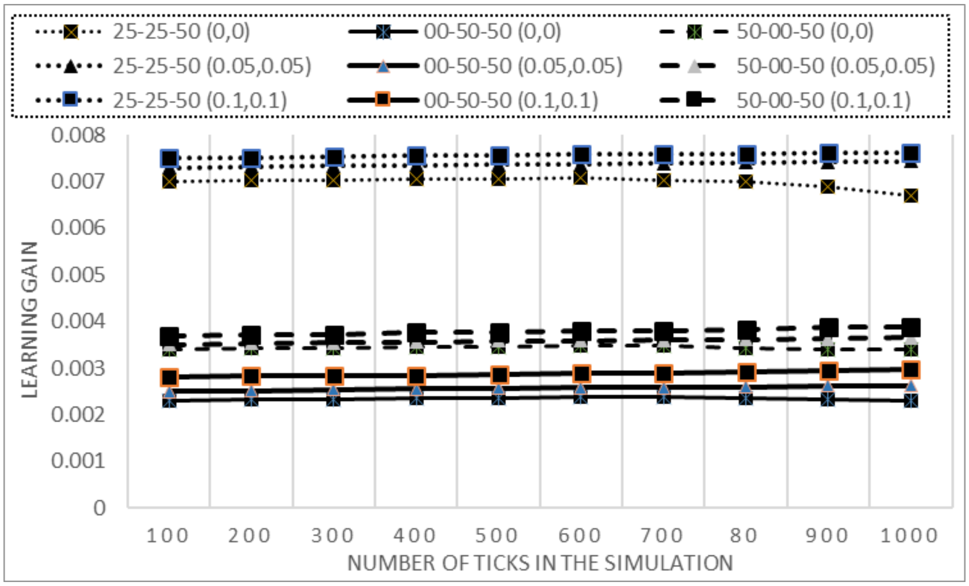


(c) TD = High

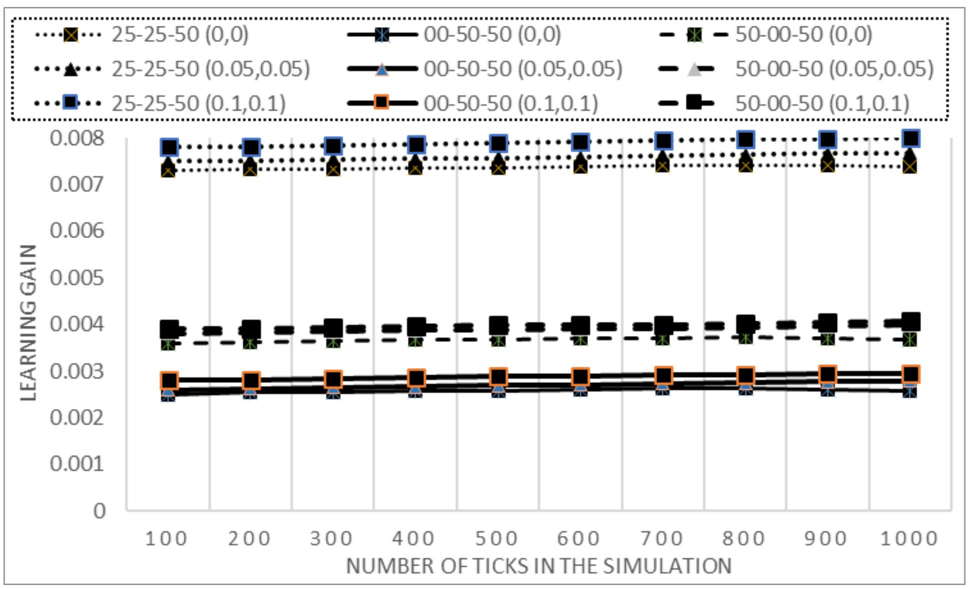
Figure A.12. Average Learning gains per task for TD = Low, Medium, and High, for AD = 25-25-50, 00-50-50, and 50-00-50, with AO = TO = 0, 0.05, and 0.1, respectively, for $p_{auctioned}$, $p < 0.001$.



(a) TD = Low

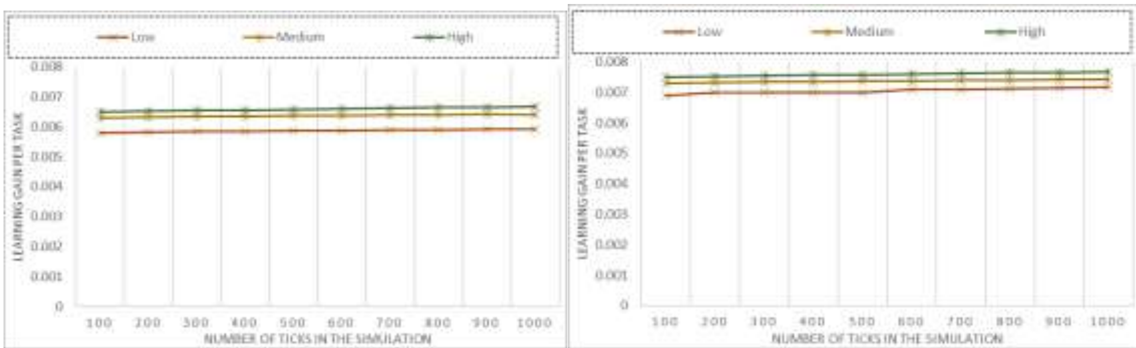


(b) TD = Medium



(c) TD = High

Figure A.13. Average Learning gains per task for TD = Low, Medium, and High, for AD = 25-25-50, 00-50-50, and 50-00-50, with AO = TO = 0, 0.05, and 0.1, respectively, for $p_{auctioned+}$, $p < 0.001$.



(a) $p_{auctioned}$ (b) $p_{auctioned+}$

Figure A.14. Average Learning gain per task for AO = TO = 0.05, AD = 25-25-50, and TD = Low, Medium, and High, $p < 0.001$.

It can be seen that the learning gains for $p_{auctioned+}$ are better than $p_{auctioned}$. As seen earlier, this is because the latter gets fewer percentage of tasks auctioned off as compared to $p_{auctioned+}$. This is because the agents tend to reduce chasing each other and instead focus on common tasks and in return get more tasks auctioned off and hence increase the learning gains per task. They show similar trends for diversity and openness:

Diversity. It can be seen that in the most diverse environment (25-25-50), the learning gain per task is the highest. This is because there is a mixture of all three types of agents, Apprentices, Generalists, and Specialists, this different level of expertise at a wide variety of tasks helps all the agents mutually learn and benefit from each other and hence get more tasks auctioned off. This also helps agents evolve which further lets them gain more expertise at a wider variety of skills and hence seeing more tasks auctioned off. 50-00-50 configuration does better than the 00-50-50 but not as good as 25-25-50. We saw a similar trend earlier from Figure A.10 as well, this trend is observed because the learning gap between Apprentices and Generalists is more ideal as compared to that between Apprentices and Specialists, thus Apprentices evolve faster and get better at a wider variety of capabilities in presence of Generalists. The reason behind this is that, as discussed in Chapter Methodology, we referred to Bandura's theory for calculating learning by observation. Based on Equation (3.3), we see that if the difference between an observer's expertise level and a performer's is too large or too small, then the observer does not benefit much from learning by observation. Thus, it is easier for an Apprentice to learn from a Generalist as compared to a Specialist, and a Generalist to learn from a Specialist. With an increase in the level of task diversity, the average learning gains increased as well since the agents had a wider variety of capabilities they learnt and could get a wider variety of tasks auctioned off as well. With the increase in the level of task diversity, the average learning gains increased as well. This is because in a task diverse environment, agents are presented with the opportunity to be able to learn a wide variety of capabilities.

Openness. The most open environment, AO = TO = 0.1 does the best followed by AO = TO = 0.05 and then the environment with no openness at all. We saw similar trends earlier with learning gains per tick as well (Figure A.10), an increase in AO and TO causes the learning gain per task to increase as well. This is because on account of newer tasks that keep coming in and new agents coming in or old ones leaving, agents always have something new to learn by doing tasks. This causes an increase in the learning gain with increasing AO and TO. It can be observed that for non-zero openness (AO = TO = 0.05 and 0.1) the learning gain curves are pretty flat. This is because openness helps the agents get acquainted with teammates with skills that were not present in the environment before, or to work on tasks that are new in the environment. This helps the agents improve their capabilities. On the contrary in case of no openness (AO = TO = 0), agents keep working with the same set of teammates with the same set of tasks. As a result, they reach a point beyond which they stop learning anything new and the learning curves thus show a downward trend.