Department of Computing

Interdependence between Agents in Multi Agent Systems

Billy Pik Lik Lau

This thesis is presented for the Degree of
Master of Philosophy (Computer Science)
of
Curtin University

January 2014

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature : ……..……

(BILLY PIK LIK LAU)

Date : January 2014

Abstract

The interaction between agents can be viewed as a relationship that allows the agents to exchange information and collaborate to solve a particular problem. The interdependence relationship between agents leads to a different formation of the interaction framework called a coalition. *Coalition formation* (CF) is one of the fundamental forms of interaction that allows the creation of coherent groups of agents based on the agents' relationships. It offers agents the capability of forming a dynamic and goal-oriented paradigm to achieve their goals effectively. However, it suffers from major drawbacks such as high communication overhead, limitations of the protocols and the high computational complexity of the algorithms. In this thesis, two aspects have been focused, i.e. to improve the cooperation between agents from the perspective of agents' interdependence relationship and existing agents' organization framework.

First, the interdependence relationships between agents based on the dependence theory have been addressed using the CF. The *transitive dependence relationships* between agents is one of our focus since indirect relationships are important for the agents to depend on each other transitively. We have developed an heuristic based algorithm that generates the coalitional value using a distributed approach and calculates the coalition cost given a constraint or budget. It has been shown that the budget provides a limit to the coalition cost during CF. The coalition earns more profit since the coalition cost has been reduced. In experiments, the agents' number

will not exponentially increase the coalition cost. Furthermore, the *validity of the relationships* is another research focus as a feasible collaboration will guarantee the profit of the individual agent in a coalition. It uses the solution derived from the 0-1 Knapsack problem and calculates the relationship based on three different types of dependence such as and, or and singleton relationships. The experimental results have shown that a smaller coalition has a higher preservation of the relationships' validity while conserving the efficiency of the communication rate.

The agents' organization framework, such as coalition, provides a sophisticated environment and protocols for agents to depend on each other for solving particular problems. However, the idle agents are not able maximize its utility because of certain issues. With this in mind, the join coalition mechanism (JCM) has been developed to help idle agents increase utility by joining the existing coalition. Two management approaches have been studied which are *macroscopic* and *microscopic* coalition. The macroscopic coalition is a leaderless coalition that uses the two-phase evaluation where public voting and trial joining are carried out. The JCM decides the proposal of idle agents that try to join an existing coalition. Experimental results based on the macroscopic coalition have shown that idle agents are able to join the coalition with the communication rate conservation up to 88.00%. We also found out the behaviour of the agents during the trial joining phase is the key to build trust. The microscopic coalition has a representative agent that serves as the leader of the organization. The two-phase joining methods are refined and involves the evaluation of the coalition leader followed by the weighted voting session by the coalition member. Experimental results show that increasing the numbers of agents intensify the competition of voting session from 5.04% to 83.19%. On the other hand, the

increment of the agree evaluation ratio made the chances for the idle agent to join the coalition raise from 0.00% to 44.36%. The JCM has successfully increased the rate of cooperation between agents by allowing the idle agents to join the existing coalition.

Acknowledgement

I would like to express my gratitude to Associate Professor Dr Ashutosh Kumar Singh and Mr. Terence Tan Peng Lian for their valuable advices and guidance throughout my research studies (Master of Philosophy, Computer Science). Their availability for discussion on my research works has further motivated me and I am truly grateful. Deeply appreciated.

In addition, I would like to thanks the Ministry of Higher Education Malaysia for the scholarship award entitled “MyBrain 15, MyMaster” and Curtin University Sarawak Campus for the “Curtin Sarawak Postgraduate Stipend Scholarship Award”. These financial aids have been a great support for funding my research projects.

My gratitude also goes to my fellow colleagues in Curtin University Sarawak Campus namely Garenth Lim, Clement Tham, Chua Shin Cheng, Chin Pin Rui and Arthur Wong for their assists on technical and non-technical support during my research studies.

All my friends and families deserve a big thank you for their precious support and encouragement throughout the past two years. In particular, special thanks to my friend - Sim Sheng Kai for his valuable time in helping me proof-reading the thesis.

Last but not least, I would like to thank God for His guidance throughout my research studies. He had led me through the hard time and had me prepared everything required for completing this thesis. All glory and honor goes to Him in the highest.

Publications

JOURNAL PAPERS

1. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, "*A Review on Dependence Graph in Social Reasoning Mechanism*", Artificial Intelligence Review, 2012, ISBN: 1573-7462 ,DOI: 10.1007/s10462-012-9371-x, pp. 1-14 (Published)

2. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, "*Forming Transitive Dependence based Coalition utilizing the Concept of Budget in Multi-agent Systems*", Journal on Artificial Intelligence Tools, 2013 (Under review, ISI indexed).

3. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, " *Join Coalition Mechanism for Enhancing Idle Agents' Utility in a Macroscopic Coalition*" Journal of Artificial Intelligence Research, 2013 (Under review, ISI indexed).

4. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, "*Enhancing Idle Agents' Utility by Joining an Existing Coalition*", Progress in Artificial Intelligence, 2014 (Under review, Scopus indexed).

5. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, "*T-DepExp: Simulating Transitive Dependence Based Coalition Formation*"

Pertanika Journal of Science and Technology (JST), 2013 (Under review, Scopus indexed).

6. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Jason Rong Kun Tan, "*A Knapsack based Dependence Relationship Validation Mechanism for Verifying Agents' Social Status*", AI & Society, 2014 (Under review, ISI indexed).

REFEREED CONFERENCE PAPERS

1. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan, "*Weighted Voting Game based Algorithm for Joining a Microscopic Coalition*", TENCON 2013, Xi'an, China, 2013 (Shortlisted for the best student paper award) (Published).

2. Ashutosh Kumar Singh, **Billy Pik Lik Lau**, Terence Peng Lian Tan, "*A New Architecture for Intelligent Agents to Join a Currently Existing Coalition*" Sixth International Conference on Advanced Computational Intelligence 2013, Hang Zhou, China, 2013 (Published).

3. **Billy Pik Lik Lau**, Ashutosh Kumar Singh, Terence Peng Lian Tan "*T-DepExp: Simulating Transitive Dependence Based Coalition Formation*" Engineering Goes Green, 7th CUTSE Conference, pp. 207-210, Miri, 2012 (Published).

Table of Contents

List of Figures

List of Tables

Abbreviations

MAS	Multi-Agent Systems
OMAS	Open Multi-Agent System
SRM	Social Reasoning Mechanism
ASIC	Architecture for Social and Individual Control
OAA	Open Agent Architecture
BDI	Belief Desire Intention
CNP	Contract Net Protocol
CF	Coalition Formation
CSG	Coalition Structure Generation
CFP	Call for Proposal
KDRVM	Knapsack based Dependence Relationship Validation Mechanism
JCM	Join Coalition Mechanism
T-DepBM	Transitive Dependence Budget Mechanism
JCR	Join Coalition Request
NIPD	n-Players Iterated Prisoner Dilemma
WVG	Weighted Voting Games
CVC	Coalitional Value Calculation
T-Dep	Transitive Dependence Relationship
Dpc	Dependence Chain
CSG	Coalition Structure Generation
CFG	Characteristic Function Game
T-DepExp	Transitive Dependence Experiment

NP	Non-Polynomial
FIFO	First In First Out
TFT	Tit for Tat
LOC	List of Coalitions
JUNG	Java Universal Network Graph
PP	Probalistic Polynomial
WVS	Weighted Voting Session
IAV	Incomplete Agree Vote
IDV	Incomplete Disagree Vote
CAV	Completed All Vote

Glossary

agt_{root}	The root agent in the dependence relationships
agt_{prt}	The parent agents that is not autonomous and depend on child agents
agt_{chd}	The agent depended by parent agent for its capabilities
Agt_{chd}	The set of the child agents in the dependence relationships
agt_{leaf}	The agent that is autonomous of its goal and end of the dependence relationships
dep_{and}	The AND dependence relationship
dep_{or}	The OR dependence relationship
dep_{1to1}	The singleton dependence relationship
j	Number of agents involved in a relation
agt_j	The agent engaged in the specific relationships
Agt_j	The set of agents engaged in the specific relationships
k	Number of goals an agent wants to achieve
agt_{join}	The agent who propose join coalition request (JCR)
agt_{lead}	The coalition leader
agt_{reg}	The coalition registration agent
agt_x	The coalition member
x	The specific number of agent in a n members coalition
β	Budget

a	Action
g	Goal
G	Set of goals
p	Plan to execute for achieving goal g
P	Set of plans
i	The possible number of agents involved in the forming of coalition
n	Number of agents
a_{out}	An action that the agent is autonomous
z	The specific coalition
Z	The set of coalition in the society
ϕ	The number of revision of the budget mechanism
${}^{n}P_{k}$	Permutation of the set n
$comp_{x}$	The forestry company
$comp_{req}$	The furniture company
$comp_{prod}$	The plywood company
G_{z}	The common goal of the coalition z
A	Set of actions an agent owns
R	Set of resources an agent owns
P_{and}	The ratio of the and dependence in the coalition
P_{or}	The ratio of the or dependence in the coalition
P_{1to1}	The ratio of the singleton dependence in the coalition
U_{i}	The estimated cost of the agent agt_{i}
σ_{i}	The priority of the agent agt_{i} ’s plan

cp_i	The agents' capabilities for fulfilling the required plan
P_a	The ratio of the dependence
U_x	Vector payoff distribution for the agent agt_x
$\sum_{i=1}^{n} R_i$	The profit of the CF
pr_h	Profit of the item
h	The number of objects in knapsack problem
W	The maximum weight in a knapsack problem
w_h	Weight of the item
σ	Weight of an agent in the dependence relationships
γ	Cost of performing an action
m_{goal}	The priority of the goal
v	Estimated profit of a coalition
φ	Maximum capacity of a root agent
c	Current capacity
μ	Communication rate
T_{sim}	The simulation time of the experiment
dep_{mix}	The mixture of the dep_{and}, dep_{or}, dep_{1to1} relationships linked to the agent agt_{prt}
P_{and}	The ratio of the AND-dependence relationships
P_{or}	The ratio of the OR-dependence relationships
P_{1to1}	The ratio of singleton relationships
C	The pay-out for both agents cooperate in PD games
D	The pay-out for both agents that defect in the PD games

E	The agents who choose to defect while the other agent cooperate in the PD games
S	The agents who choose to cooperate while other agent defect in the PD games
T_{α}	The trust metric of an agent
Q_{agt}	The strategic of an agent in the NIPD game
P_{s1}	The ratio of "desperate" agents in the society
P_{s2}	The ratio of "grumpy" agents in the society
P_{s3}	The ratio of "imitator" agents in the society
P_{s4}	The ratio of "avenger" agents in the society
P_{s5}	The ratio of "uncertain" agents in the society
$\sum_{u} P_{su}$	The ratio of the agents in the coalition
u	The number of strategies
d	The number of iteration for NIPD games
RE	Real number
l	The numbers of the game both agent and coalition want to play initially.
H	The winning count of NIPD games
$\overline{H}$	The losing count of NIPD games
T_{ω}	The threshold for the trust factor.
M'	The possible interaction between agents in the coalition
m	The number of communications between agent in session
$\Re$	The number of coalition in the LOC
$\Im$	The number of coalition registered in the databases

sc	The total score of the NIPD games
$\aleph$	The average score of the NIPD games
E_{μ}	The unutilized the connections
Θ	The agents' weighted vote
Θ_{no}	The agents' weighted vote on disagreement during voting session
Θ_{yes}	The agents' weighted vote on agreement during voting session
C_e	The criteria to evaluate a decision which allocated for weighted vote
con_x	The agents' contribution in a society
y	The number of criteria to be evaluated
Ω	The quota for the weighted voting session
Z'	The reformed coalition after the agent agt_{join} joined
$CR_{\Theta yes}$	The agree evaluation ratio of the agent
ω_{Θ}	Voting percentage

Chapter 1 Introduction

Multi-Agent Systems (MAS) is a group of intelligent agents work together to solve a large scale and complicated problem. In real life applications, MAS can be found in the application of autonomous robots, ad hoc network, cluster computing and global network design. As the technology grows, MAS become a preferable paradigm for handling complex systems. One may argue that a singular entity without communication is more efficient than the multiple entities where the communication between agents and computational resources are expensive. However, these limitations have been mitigated through the rapid growth of technology that boosts the research work on various applications using MAS architecture. According to Wooldridge (2009), the MAS can be defined as the following:

> *"Multi agent systems are systems composed of multiple interacting computing elements, known as agents. Agents are computer systems with two important capabilities. First, they are at least to some extent capable of autonomous action - of deciding for themselves what they need to do in order to satisfy their design objectives. Second, they are capable of interacting with other agents - not simply by exchanging data, but by engaging in analogues of the kind of social activity that we all engage in every day of our lives: cooperation, coordination, negotiation, and the like."*

The main idea behind the MAS is due to the limited capabilities of a single agent, that agent will communicate with other agents in order to achieve its goals. An

agent's intention of communicating other agents for required capabilities are developed through the concept of interdependence. There are several critical types of issue in the MAS such as distributed planning between agents, coordination of agents for tasks sharing and resolving conflicts (Durfee, 2006; Sycara, 1998; Vokrinek, Komenda, & Pechoucek, 2011; Wooldridge, 2009). The following Figure 1.1 shows the categorization of the MAS of our works which includes managing resources, communication and organization:

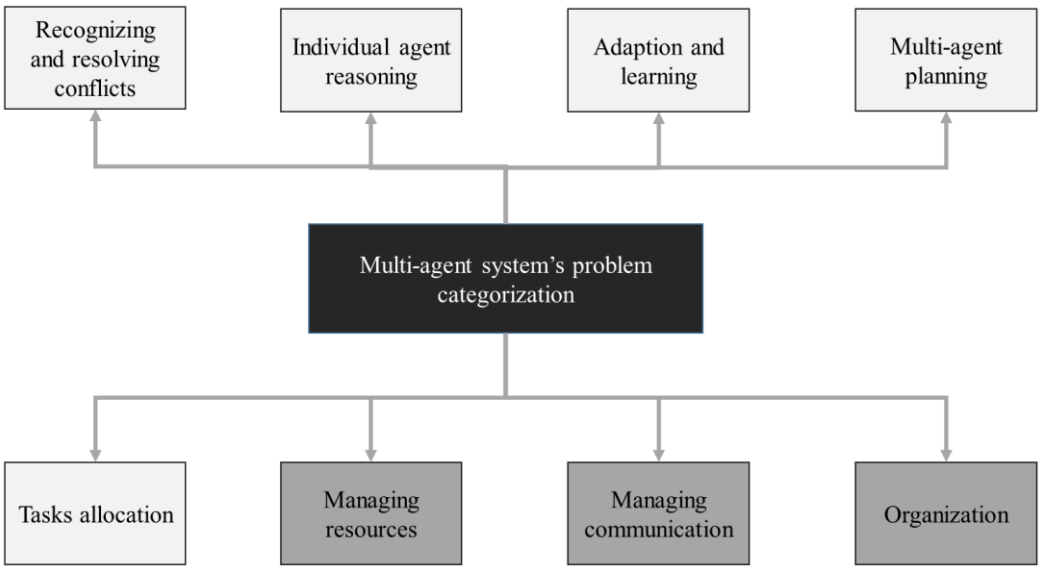


Figure 1.1 The categorization of the MAS problem

Today, the major challenge in MAS research are the communication and resource management between agents for large scale of system networks. The communication between agents composes of interaction that enables cooperation, coordination and negotiation. Communication overhead can be generated over time along when agents communicate with each other. It is the extra information that consumes resources to maintain agents' communication such as acknowledgement and signalling data. The agent can be affected by the frequent communication overhead which causes interruption. It is important to reduce the communication overhead by managing the

resources efficiently.

One of the ultimate goals is to increase the efficiency of the open multi-agent system (OMAS) where agents work in an extremely dynamic and cross interoperability environment. The communication between agents is crucial when performing certain actions and tasks to ensure interoperability and coherence among themselves (d'Inverno, Luck, Wooldridge, & others, 1997; J. S. Sichman & Demazeau, 2001). Due to different agents' capabilities, they will need to depend on other agents for specific actions or resources to reach their goals. This scenario is known as a heterogeneous system.

As a socially intelligent agent, it will need to coordinate with other agents in the society to initiate the collaboration in order to achieve its goal efficiently. Intense communication between agents can cause undesirable communication congestion between agents in the society. In order to mitigate the communication flow and overhead, *social reasoning mechanism* (SRM) has been formed (J. S. Sichman, Conte, Demazeau, & Castelfranchi, 1998). SRM is a mechanism to indentify the agent's dependence relationships for collaboration's optimization. The *Dependence theory* was originally presented by Sichman et al. (J. S. Sichman, 1998; J. S. Sichman & Conte, 2002; J. S. Sichman & Demazeau, 2001) to address the need of an agent to achieve a goal by depending on others. Dependence network and graph model have been developed to further visualize the agents' relationship. The fundamental structure of the dependence graph is based on the external description of the agents' architecture which is represented using graph theory (Bondy & Murty, 1976). The agents' external descriptions contain the data structure of agents' goals,

plans, actions and resources. Figure 1.2 shows the implementation of the SRM based on the agent model, architecture for social and individual control (ASIC) which was initially developed by Oliveier Boissier and Demazeau (1994).

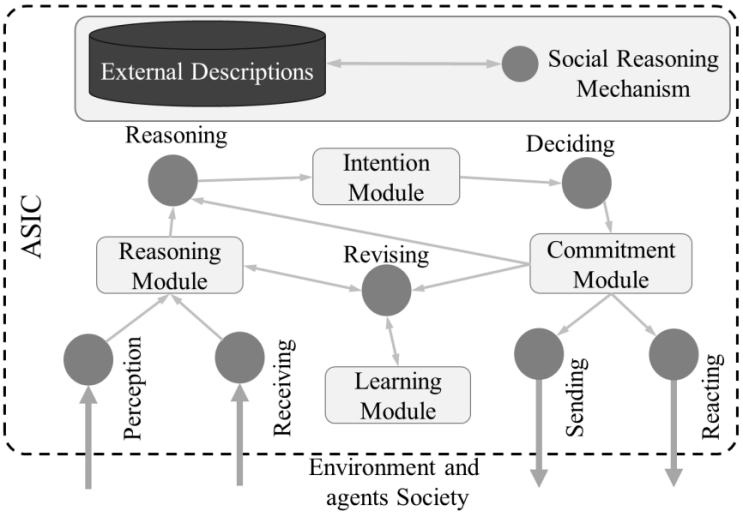


Figure 1.2 Agent model in ASIC model (Oliveier Boissier & Demazeau, 1994)

Several architectures have been proposed to model the social interaction such as social believable architecture (Guye-Vuillème & Thalmann, 2000) and open agent architecture (OAA) (Martin, Cheyer, & Moran, 1999). OAA proposes the concept of creating agents that are capable of communicating with legacy agents by implementing several languages interpretation techniques. It boosts the agents' social capability by reducing the cooperation gap. On the other hand, the OAA emphasizes the interoperability between agents for their interaction. To generalize the design of a complex software, Wooldridge et al. (Wooldridge, Jennings, & Kinny, 2000; Zambonelli, Jennings, & Wooldridge, 2003) have proposed a Gaia methodology to design an agent based computing software following certain guidelines. The reasoning process takes information from the perception module and the communication data sent by other agents to perform reasoning. The intention module helps an agent in decision making based on the reasoning of the current state. The commitment module reacts to the decision made by an agent accordingly. The ASIC

model follows the "belief, desire and intention" (BDI) framework (Wooldridge, 2009) and have been implemented into the SRM. The dependence theory relies on a few assumptions: (1) benevolence principles, (2) sincerity principles, (3) auto-knowledge principles and (4) consistence principles (J. S. Sichman et al., 1998). One of the significant contributions of SRM is the reduction of communication flow between agents (J. S. Sichman & Demazeau, 2001). Through the SRM, agents can identify its relations and choose an agent with the highest profit to cooperate. The identification of the dependence relationships is essential to significantly reduce the communication overhead between agents. Some well-known protocols such as contract net protocol (CNP) (Smith, 1980) has benefited from such identification. However, SRM has some unsolved puzzles such as validity of the dependence relationships, social degree and real world implementation. The social degree accounts the number of relations an agent engaged to analyse the intensity of cooperation.

Various organizational framework have been surveyed by Horling and Lesser (2004) as illustrated in Figure 1.3:

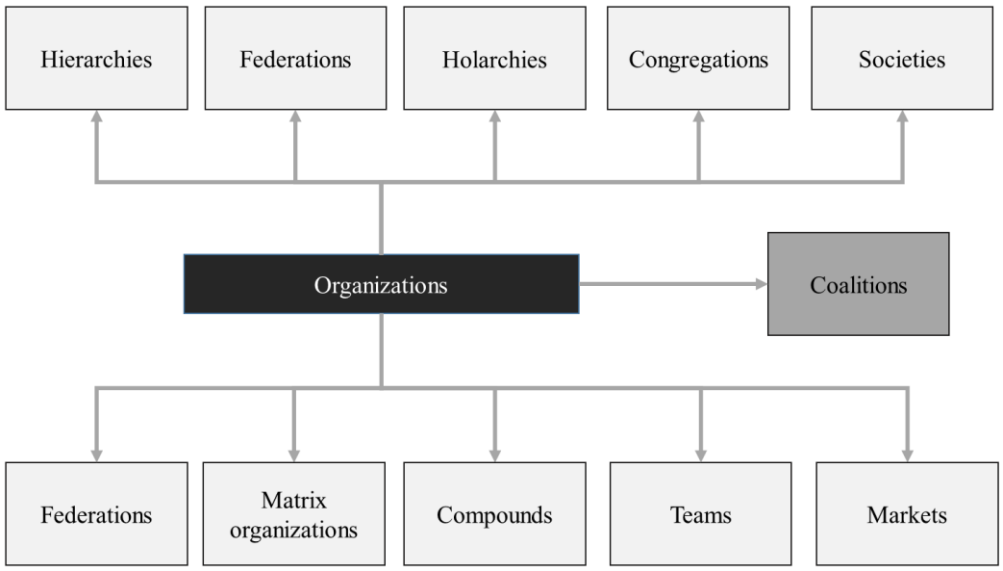


Figure 1.3 The classification of the organizational paradigms in MAS

The organizational framework focused in this thesis is the *coalition* and it has a few advantages over other organizational frameworks. A coalition advocates a goal-directed paradigm to form a short-lived organization. In addition, it allows agents to adapt the dynamic environment while enforcing cooperative strategy to solve the problem efficiently. The coalition has been studied widely by researchers (Adel, Rajabi, & Reza, 2010; Boella, Sauro, & Torre, 2006; Chalkiadakis, Markakis, & Jennings, 2012; Kraus, Shehory, & Taase, 2003; Nathan & Luck, 2003; Sandhlom & Lesser, 1997; Sandholm, Larson, Andersson, Shehory, & Tohmé, 1999; Wojciech & Kwasnicka, 2008) because of its significance in a dynamic environment and effective agents collboration. The forming of the coalition, namely *coalition formation* (CF) originates from game theory (Osborne & Rubinstein, 1994) where it was applied to study the partition of agents into possible groups. Due to the large search space, the CF problem normally is an intractable problem where it falls under *non-polynomial* (NP)-hard problems. The CF composes of three core activities, i.e. *coalitional value calculation* (CVC), *coalition structure generation* (CSG) and *payoff distribution*. The details of the three activities will be further explained in the next chapter.

In normal circumstances, agents will attempt to gain profit by cooperating with other agents. Often, they received invitations from other agents before the coalition is formed. Once an agent has accepted the invitation, it will become part of the coalition and contribute its capabilities to achieve common or shared goal. If it does not receive invitations from others, the agent is still able to form a coalition by sending a call for proposal (CFP) to others using CNP. If agents agree proposal of CF, they will cooperate and try to achieve the common goals of the coalition. What if an idle agent receives any invitation or do not have the required budget to form a

coalition? According to our literature review, there is related works or mechanism on joining existing coalition proposed before. Hence, we have carried out in depth investigation to study the possibility for idle agents to join existing coalitions.

1.1 RESEARCH CONTRIBUTIONS

The ultimate goal of this research work is to improve the cooperation between agents in the MAS through their interdependence relationship. To this end, an efficient protocol or mechanism is developed to increase agents' rate of cooperation. The following are the significance of our research:

- Develop the CVC and CSG based on agents' dependence relationship.

The coalition value serves as an important numerical metric for agents to determine individual profit in a coalition. There are currently two methodologies have been developed to address the CVC which are Shehory and Kraus (1998) and Rahwan and Jennings (2007)'s algorithms. However, they are complicated and heavily rely on the knowledge of every agent in the coalition. Our methodology uses the technique of passing information through the dependence relationships as every agent is able to perform the calculation of estimated cost and the validity of CF locally. It uses the concept of coalition budget that serves as maximum threshold for the total coalition cost.

- Verification of the dependence relationships' validity

The dependence relationships play an important role to determine the computational complexity of a coalition through their interaction. The validity of relationship is important to ensure the profit of individual agents. Through the development of the

knapsack based dependence relationship validation mechanism (KDRVM), the agents are able to verify their relationship with a constraint. It proposes the concept of 0-1 knapsack and the individual profit to check its validity. If the coalition does not ensure individual agents' profit, then the coalition is considered as not feasible. Hence, agents will suffer from poverty and lead to coalition's profit lost.

- Allows idle agents to join an existing microscopic and macroscopic coalition.

Idle agents are bounded by several limitations such as not receiving the CFP and having limited budget to form a coalition. Hence, we introduce the join coalition mechanism (JCM) to help agents join existing coalitions. We studied two management approaches studied which are macroscopic and microscopic coalition. Two different joining approaches for idle agents have been implemented to both macroscopic and microscopic coalition which are n-players prisoner dilemma (NIPD) games and weighted voting games (WVG) respectively. One of the non-cooperative game approach, prisoner dilemma has been used as one of the trial joining phase for macroscopic coalition. On the other hand, the weighted voting game has been implemented to address inequality of agents in the microscopic coalition.

1.2 THESIS STRUCTURE

In the remainder of thesis, we present the preliminaries and then describe the literature review as well as backgrounds of the proposed solution. Subsequently, the algorithm for calculating the coalitional value and forming coalition structure are explained. We later propose the algorithm to verify the agents' relationships' validity to ensure profit of coalition while minimising the communication effort. The JCM

has been presented to help idle agents join existing coalition based on different perspectives. This is achieved through the rest of the remaining chapters which are structured as following:

- In Chapter 2, the agents' architecture is modified to suit our proposed algorithms is presented. After that, agents' perspective and agents' roles are explained. The mathematical notations for the CF are presented which includes elements such as coalition cost, budget and profit. Two different type of coalition management are addressed as well. Lastly, the dependence relationships between agents in the coalition are described.

- In Chapter 3, we reviewed and discussed the dependence graph which is also known as interdependence graph in some author's work (Conte & Sichman, 2002; J. S. Sichman & Conte, 2002). The dependence network and graph and the CF which serve as the foundation of our work are explained. The backgrounds, type of managements and current issues are discussed and further analysed.

- In Chapter 4, the transitive dependence budget mechanism (T-DepBM) has been proposed to resolve individual agent's cost and CSG using the concept of budget. The payment configuration calculation of the current transitive dependence relationships is incomplete where our proposed mechanism ensures the coalition's profit through the feasibility of budget. Subsequently, the simulations have been carried out and the results have been analysed. However, the CVC problems remain open even with T-DepBM.

- In Chapter 5, we have discussed about the stability of a coalition due to the CVC problem arised in Chapter 4. We have proposed the KDRVM which validates the dependence relationships based on the analogy of the bounded 0-1 knapsack problem. It ensures every dependence relationships in the coalition reaches a stable state where profit is guaranteed for every valid relationship. The experiments regarding the agents' number and relationships' types have been conducted to verify the effectiveness of the proposed mechanism. After fixing the calculation of individual agent's budget and profit, we will start to increase agents' utility by introducing external agents to join existing coalition.

- In Chapter 6, the JCM has been proposed for providing idle agents the ability to join an existing macroscopic coalition to maximize their utility. The JCM uses two-phase evaluation which includes join coalition request (JCR) and n-players iterated prisoner dilemma (NIPD) games as a trial joining interaction. The experiments in this chapter study the possibility of an agent that joins the macroscopic coalition and behaviour of agents. By understanding the management of macroscopic paradigm, our next research focus shift to microscopic coalition where agents are treated uneven in a coalition.

- In Chapter 7, the JCM based on the weighted voting game (WVG) is extended to help idle agents to join an existing microscopic coalition. It includes distributed algorithms based on the agents' roles with the evaluation

of the goals, budget and mutual trust of joining agent. The simulations have been carried out using the parameters such as agents' agree ratio, agents' number and WVG's quota.

- Finally, the Chapter 8 concludes the research findings on the experimental or simulation results obtained through the proposed mechanisms. Furthermore, we have outlined the possible future works that further optimize and enhance the current proposed mechanisms.

Chapter 2 Preliminaries

In this chapter, the fundamental notion of agents' architecture, agents' roles, coalition notation and dependence relationships are presented. First, we describe the agents' architecture followed by the agents' representation. Next, the agents' roles during the joining process of the idle agents are explained followed by the CF's notation. The last subsection consists of the dependence relationships built by agents during collaboration in the coalition.

2.1 AGENTS' ARCHITECTURE

The agent architecture is essential to determine the agents' core functions such as (1) reasoning, (2) deciding, (3) communication and (4) revising their belief. We have proposed a modified architecture (based on ASIC (Olivier Boissier & Demazeau, 1996)) that allows agents to work in an organization using the BDI architecture. Also, this architecture has been simulated in the T-DepBM, KDRVM and JCM for simulating the CF among a group of agents.

Our proposed agents' architecture in Figure 2.1 has the features of helping an agent to form the coalition by itself as well as joining an existing coalition. We have modified ASIC by including trust, vote and budget mechanisms. The modified ASIC uses agents' external descriptions as a mean of obtaining knowledge for certain level of decision making. The Figure 2.1 shows the modified version of ASIC architecture.

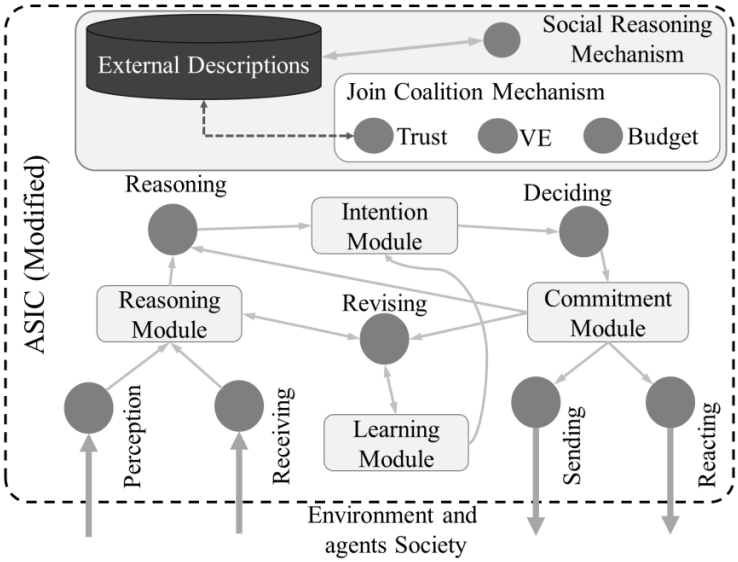


Figure 2.1 The modified version of ASIC architecture

2.2 AGENT'S REPRESENTATION

The representation of the agents' interaction is based on their relationships with others and their current tasks. It covers the aspects of depending agent (known as depender) and agent being depended (known as dependee). The Figure 2.2 illustrates the example of each agent's representation of their relationships in graph form.

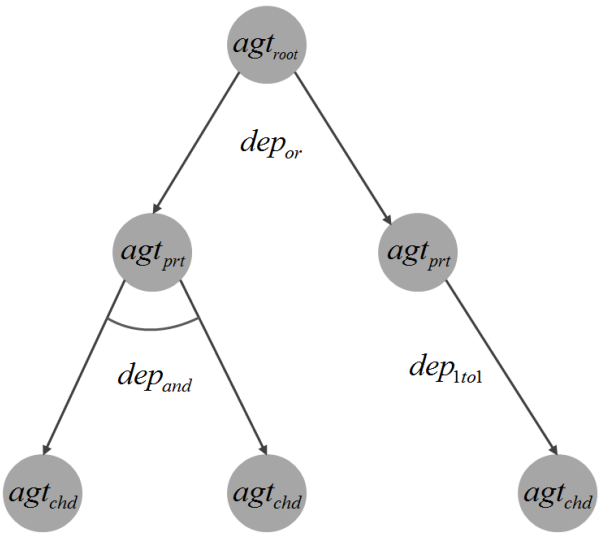


Figure 2.2 The agents' representation in their relationship

There are four types of agents' representation role in the relationship which are root agent (agt_{root}), child agent (agt_{chd}), parent agent (agt_{prt}) and leaf agent (agt_{leaf}). The following sub-sections are the in-depth explanation of each agent's representation in the relationship.

2.2.1 Root Agent

The root agent agt_{root} is the agent on top of all dependence relationships in a coalition. The root agent agt_{root} is also a parent agent since it has some child agents where it is not autonomous of its capability. An agent agt_{root} may have j child agents to depend on in order to achieve its goal g_{root}. The characteristic of root agent agt_{root} includes it does not have the parent agent and it is also not a child agent. In a microscopic coalition, the agent agt_{root} can be viewed as a leader in the organization.

2.2.2 Child Agent

The agent agt_{chd} is dependent on the parent agent agt_{prt} for its capabilities including actions and resources. The number of child agent j in a dependence relationship is based on the goals and plans of the parent agent agt_{prt}. A child agent agt_{chd} can be the parent agent of others if it is not autonomous of its capabilities to achieve its goals. In the remainder of this thesis, the set of child agents is denoted as Agt_{chd} for representing the group of agent agt_{chd} involved in a relationship.

2.2.3 Parent Agent

A parent agent agt_{prt} is the agent that depends on child agents for achieving its goals.

The parent agent is not autonomous of its capabilities which cause the dependence relationship to occur. It is denoted as $(agt_{root} \in Agt_{prt}) \wedge (agt_{root} \notin Agt_{chd})$.

2.2.4 Leaf Agent

The leaf agent agt_{leaf} is a subset of the child agent Agt_{chd}. It is denoted as $(Agt_{leaf} \in Agt_{chd}) \wedge (Agt_{leaf} \notin Agt_{prt})$. The main difference between an agent agt_{chd} and agt_{leaf} is the agent agt_{leaf} does not have child agents which indicates the end of the dependence graph / relationships. It means the leaf agent is autonomous of its capabilities. In addition, it means the goal of the agent agt_{prt}'s goal has been fulfilled.

2.3 AGENTS' ROLES

Each agent has different tasks corresponding to their activities assigned by root agent agt_{root}. We have categorized agents by their role to indicate their current state in a coalition. The agent's current state is important to generating differential level of decision making among coalition members. There are four types of agent's roles identified which are joining agent (agt_{join}), coalition representative (agt_{lead}), coalition registration agent (agt_{reg}) and coalition member (agt_x).

2.3.1 The Joining Agent

The joining agent agt_{join} is the agent with the intention of joining an existing coalition in order to maximize its utility. The agent agt_{join} has a set of goals that it wants to achieve and are denoted as G_{join}. It tries to join the existing coalition if the

following conditions are satisfied:

Condition 2.1: The agent agt_{join} and the targeted coalition share a set of common goals.

The first condition shows the agent agt_{join} shares a common goal with the targeted coalition z. This implies the agent agt_{join} is able to achieve its and coalition goals through the coalition joining process. The agent agt_{join}'s common goal g_{join} and the targeted coalition's goal are denoted as following:

$$\exists g_{join} \in G_z \tag{2.1}$$

The equation above denotes agent agt_{join}'s goal exists as a subset of the coalition z's goal G_z.

Condition 2.2: The agent agt_{join} does not have the required budget to form its coalition or receive any invitation from other coalitions.

Every realistic agent model has a limited budget to be utilized or shared with others. In our proposed model, the agent agt_{join} has insufficient resources to form its own coalition. Hence, it will attempt to seek for any available coalition invitation. What if the agent agt_{join} does not receive any invitation? The agent agt_{join} will be idle and unable to maximize its utility. This scenario can be denoted as:

$$\left(\beta_{join} < \beta_{\min}\right) \cup \left(!\exists \sum_{b=1}^{\Re} inv_b \right) \tag{2.2}$$

The β_{join} represents the agent agt_{join}'s budget and $\beta_{\min}$ is the budget required for forming the coalition. The minimum budget required is capped at 25% of the average coalition by the agent agt_{reg} and announced to all agents in the same ecosystem. The inv_b denotes the invitation from any coalition given the total coalitions existed which is $\Re$.

Condition 2.3: The agent agt_{join} is not autonomous on its own goal.

The agent agt_{join} needs to depend on other agents in order to achieve its goal because it does not have the required capabilities. The Condition 2.1 and Condition 2.2 must be satisfied before the agent agt_{join} can propose JCR to the targeted coalition. If both conditions are not satisfied, it denotes the agent agt_{join} either can form its own coalition or accepts the invitation from others.

$$!\exists (A_{join} \cup R_{join}) \in P_{G_{join}} \tag{2.3}$$

The A_{join} and R_{join} denotes the actions and resources that an agent agt_{join} possessed. The $P_{G_{join}}$ is the plan of the agent agt_{join} to achieve the goal g_{join}. The goals and plans is based on the other agents' external descriptions which consists of actions and resources.

2.3.2 The Coalition Representative

The agent agt_{lead} is the representative agent of the coalition that manages the organization's affair such as tasks distribution, planning, scheduling and etc. In other words, it serves as the leader of the coalition. It holds the responsibility for the planning and scheduling tasks for the members. In the remainder of thesis, the agent with the highest contribution is denoted as the leader of the coalition. In addition, the agent agt_{lead} is enfranchised to vote in the proposed JCM.

2.3.3 The Coalition Registration Agent

The coalition registration agent (CRA) agt_{reg} is the middle-man agent and has the goal to maximize its profit through the forming of coalition. It also serves as the intermediary between external agent and agent agt_{join}'s targeted coalition during the process of proposing JCR.

2.3.4 The Coalition Members

The set of agents Agt_x denotes the member of the coalition that consists up to n number of members. The enfranchisement indicates each agent agt_x has the "*right to vote*" when a voting session is triggered. The $agt_{x\in\{1,2,\ldots,n\}}$ represents one of the coalition member from the set of agents Agt_x that are written in a smaller caps. Besides, the coalition representative agt_{lead} is a subset of coalition members Agt_x where its contribution is higher than other agents.

2.4 COALITION FORMATION

The coalition has been chosen as the organization for the agents to form cooperation

because it offers more dynamic and goal oriented group formalization. It ensures agents are able to work towards common goals while maximizing their individual utility. The further details of other organization framework can be found at Horling and Lesser (2004)'s survey. The coalition paradigm originated from game theory and aim to solve the problem of forming a team of players with a unique value assigned. It is also known as coalitional game with transferable utility (Adel et al., 2010) but not all of coalitional game have the transferable utility feature.

In our proposed model, we allow the singleton coalition where it consists of one member. The singleton coalition shows an agent is autonomous of its capabilities and not require to depend on other agents. The general notation for our proposed coalition paradigm is denoted as three tuples in the following equation:

$$z = < N, \sum_{i=1}^{n} U_i, G > \tag{2.4}$$

where N is the set of agents that consists of $\{1,2,...,n\}$ members. The characteristic function $\sum_{i=1}^{n} U_i$ is the real number that associates a value to the group as a whole. The individual value is denoted as payoff vector U_x. The set of goals G is the common goals among the coalition members Agt_x share. It can be denoted as following:

$$\{g_1, g_2, ... g_j\} \in G_z \tag{2.5}$$

where the g represents the goal of individual agent in the coalition and the j

represents the number of common goal a coalition z holds.

Assumption 2.1*:* Agents Agt_x adapts the sincerity principle when collaborate with other agents. This suggests the agents agt_x trust each other once the initial cooperation between agents has been formed.

Assumption 2.2: The communication channel between agents is assumed to be noiseless and lossless. There is no disturbance in the communication channel and no communication lost during the agent agt_{join}'s proposal of JCR to the targeted coalition.

First, we define the characteristic function of the coalition that calculates the total coalition cost. Then the budget of coalition will be presented. The profit of coalition is introduced to calculate the gain of coalition.

Definition 2.1: The total coalition cost $\sum_{i=1}^{n} U_i$ is the estimated total utility of every agent agt_i during the CF.

This notation denotes the cost of the agent agt_i to perform required capabilities for the parent agent agt_{prt} to achieve goal g_{prt}. The estimated cost of the agent agt_i is calculated using the following notation:

$$U_i = \sigma_i \times cp_i \tag{2.6}$$

where the U_i denotes the agent agt_i's estimated cost and the σ_i is the priority of the plan. The priority of the plan shows the grading of agt_i's tasks importance for achieving the goal G_z and is bound by the scale of $1 \le \sigma_i \le 10$. The σ_i denotes the grading of the required agt_i's capabilities based on its capacity with the scale of $1 \le cp_i \le 10$. The grading of agents' capabilities is benchmarked based upon the computational resources required. The computational percentage usage has been converted to the scale of $1 \le cp_i \le 10$. By substituting the value of plan's priority and grading of the capabilities, the utility for an agent agt_i has the minimum and maximum value of $1 \le U_i \le 100$. The following Example 2.1 demonstrates the calculation of the agent agt_i estimated cost.

Example 2.1: Consider agent agt_1 has the plan P_1 to achieve goal G_1. The plan's priority for the goal G_1 is graded as $\sigma_1 = 9$ and it has a significant impact to achieve an individual goal. The agent agt_1 has estimated that a computation power of 50% is required to execute the plan P_1. By dividing the computational complexity 50%/10, the grading systems will output $cp_1 = 5$. The agent agt_1 will have the estimated cost of $U_1 = 9 \times 5$ and output is $U_1 = 45$.

Definition 2.2: The budget β_{root} is the threshold or quota for the coalition estimated cost.

The budget β_{root} is the fund of the agent agt_{lead} allocated for forming the coalition. It serves as the maximum cost an agent agt_{lead} is willing to pay to achieve its goal.

Condition 2.4: The estimated total coalition cost, $\sum_{i=1}^{n} U_i$ must be lower than the budget β_{root}

The total estimated cost $\sum_{i=1}^{n} U_i$ represents the projected cost for the set of agents $\{agt_1, agt_2, ..., agt_n\}$'s capabilities to execute the common goals. It will be lower or equal to the agent agt_{lead}'s pre-defined budget β_{root} as shown in the following notation:

$$\sum_{i=1}^{n} U_i \leq \beta_{root} \tag{2.7}$$

Definition 2.3: The coalition's profit $\sum_{i=1}^{n} R_i$ denotes the utility earned by the coalition after deducting the operation cost.

The profit of the agent agt_i is calculated by using the actual cost U'_i to deduct the estimated cost U_i. It is denoted as the following notation:

$$\sum_{i=1}^{n} R_i = \beta_{root} - \sum_{i=1}^{n} U'_i \tag{2.8}$$

where the R_i denotes profit earned for individual agent through the CF and the U_i is the projected cost by the agent agt_i. The U'_i denotes the actual operation cost after the T-Dep based CF has been formed.

There are two types of coalition management approaches studied in this thesis. It can

be categorized into microscopic and macroscopic coalition based on the presence of the coalition leader. The characteristic of macroscopic and microscopic based coalition have been detailed in the following subsections:

2.4.1 Macroscopic Coalition

A macroscopic coalition denotes an organization that does not have a leader in charge of the group's tasks planning and allocating. To increase the fairness of the public voting inside a coalition, CRA agt_{reg} is introduced. It serves as an intermediary for the coalition members while performing voting session. The decision making between agents will be faster and fairer due to the transparent results among the coalition members. Figure 2.3 shows an example of a macroscopic coalition with five agents.

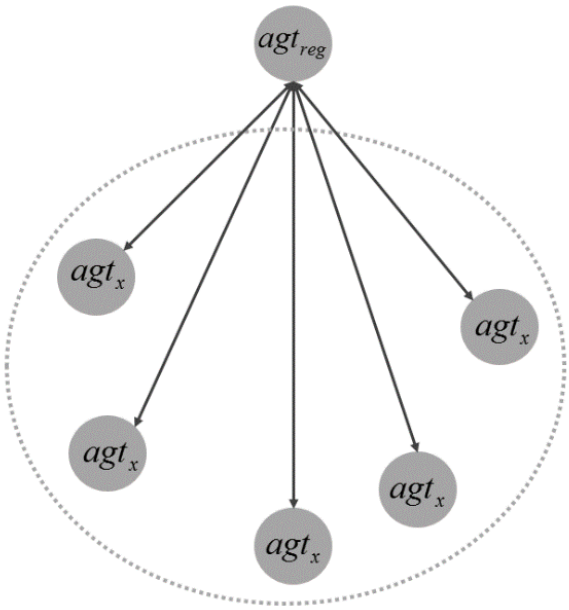


Figure 2.3 Example of macroscopic coalition with five agents

2.4.2 Microscopic Coalition

A microscopic coalition denotes an organization which has a leader that manages the

group affair. The coalition leader agt_{lead} is responsible for the planning, scheduling and allocation of tasks of a coalition. It is based on a two level hierarchical design where all the coalition members are under the agent agt_{lead}. Figure 2.4 shows the example of a microscopic coalition with five agents.

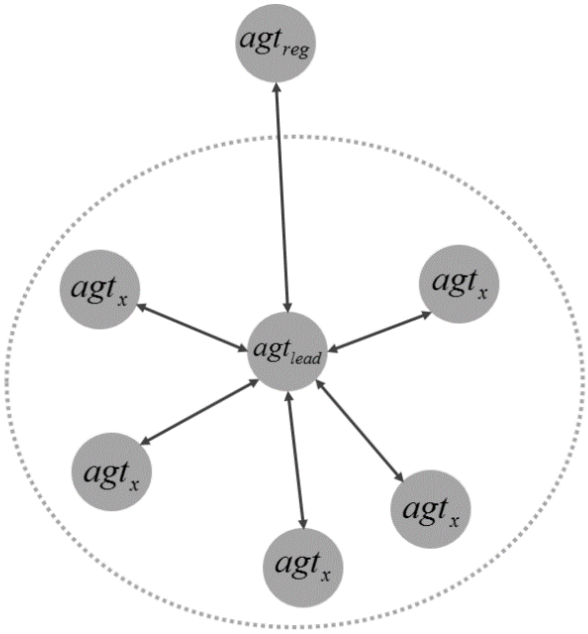


Figure 2.4 Example of microscopic coalition with five agents

In Chapter 7, we will implement the JCM into a democracy based microscopic coalition where the agent agt_{lead} do not have complete authority on deciding the organization affairs. Every decision making has to be performed through public voting by every coalition members.

2.5 DEPENDENCE RELATIONSHIPS

There are three types of the dependence relationships to-date, which are Singleton, OR and AND Dependence. It is originally presented in the SRM (Gaspar & Morgado, 2000; J. S. Sichman et al., 1998) and we have chosen it as the foundation of denoting agents' relationship in the coalition. The following subsection shows the

characteristic of three relationships respectively:

2.5.1 Singleton-Dependence

The singleton relationship dep_{1to1} relationship involves two agents and formed an one to one dependence. The parent agent agt_{prt} only depends on an agent agt_j for the particular actions to achieve its goal. Equation 2.9 denotes the notation of the dep_{1to1}:

$$dep_{1to1}(agt_{prt}, agt_j, p_{qk}, a_m) = (\exists g_k \in G(agt_{prt}) \neg a_{aut}(agt_{prt}, g_k, p_{qk}))$$
$$\wedge (agt_j \in Agt_j basic\_dep(agt_{prt}, agt_j, g_k, p_{qk}, a_m))$$
$$\wedge (\neg \exists Agt_m Agt_j \subset Agt_m dep_{or}(agt_{prt}, Agt_m, g_k, p_{qk}, a_m)) \quad (2.9)$$

Briefly, the Equation 2.9 shows that the agent agt_{prt} is not autonomous of its goal g_k and is required to depend on the only one agent agt_j for the required action a_{prt}. The following Figure 2.5 illustrates a singleton relationship dep_{1to1}.

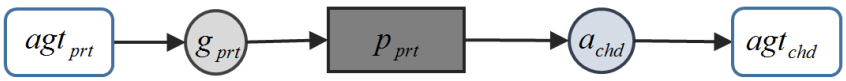


Figure 2.5 The singleton relationship

The singleton dependence only consists of one parent agent agt_{prt} and a child agent agt_{chd} under the plan P_{prt} for achieving the goal g_{prt}. The child agent agt_{chd} will supply its action a_{chd} to the parent agent agt_{prt} to complete the dependence relationship.

2.5.2 OR-Dependence

The OR-Dependence relationship dep_{or} denotes a parent agent agt_{prt} chooses a partner from a set of agents Agt_j to depend on. The dep_{or} occurs when there is a set of agents Agt_j which can secure the agent agt_{prt}'s goal g_k of the same action. The following equation denotes the OR-Dependence relationship dep_{or}:

$$\begin{aligned} & dep_{or}(agt_{prt}, Agt_j, g_k, p_{qk}, a_m) = (\exists g_k \in G(agt_{root}) \neg a_{aut}(agt_{prt}, g_k, p_{qk}) \wedge |Agt_j| > 1) \\ & \wedge (\forall agt_l \in Agt_j basic\_dep(agt_{prt}, agt_l, g_k, p_{qk}, a_m)) \\ & \wedge (\neg \exists Agt_m Agt_j \subset Agt_m dep_{or}(agt_{prt}, Agt_m, g_k, p_{qk}, a_m)) \end{aligned} \tag{2.10}$$

In short, the Equation 2.10 shows the agent agt_{prt} depends on the set of agents Agt_j to perform action a_m with alternate choices. The alternate choices are the possible set of agents Agt_j for the agent agt_{prt} to depend on. There are at least two agents involved with the action a_m in the dependence relationship to form the dep_{or} otherwise singleton dependence dep_{1to1} will occur. The last line of Equation 2.10's recursive definition ensures the Agt_j is a maximal set. The Figure 2.6 illustrates the OR-Dependence relationships dep_{or}.

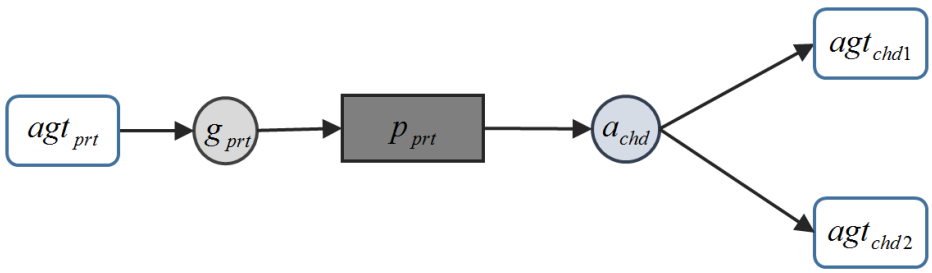


Figure 2.6 The OR-dependence relationship

The OR-Dependence relationship shows the agent agt_{prt} has the options to choose between child agents $\{ agt_{chd1}, agt_{chd2} \}$ for the action a_{chd} to achieve goal .

2.5.3 AND-Dependence

The AND-Dependence dep_{and} denotes an agent agt_{prt} that depends on every agent belonging to the set of agents Agt_j to achieve its goal g_{root}. The AND-Dependence dep_{and} denotes the agent agt_{prt} must either depend on series agents Agt_j's capabilities in order to achieve the goal g_j or search for alternate relationships. The agent agt_{prt} does not have option of choosing particular agent agt_j since every agent in the Agt_j is essential for achieving the goal. The following Equation 2.11 denotes the AND-Dependence dep_{and}:

$$
\begin{aligned}
& dep_{and}(agt_{prt}, Agt_j, g_k, p_{qk}) = (\exists g_k \in G(agt_{root}) \neg a_{aut}(agt_{prt}, g_k, p_{qk})) \\
& \wedge (\forall l_m(p_{qk}) \in I(p_{qk}) a_m \in A_n(agt_{prt}, g_k, p_{qk})(\exists! Agt_k \in AGT_j \mid agt_k \mid)) \\
& = (!\exists agt_l \in Agt_k basic\_dep(agt_{prt}, agt_l, g_k, p_{qk}, a_m)) \\
& \quad \vee (\exists Agt_k \in AGT_j dep_{or}(agt_{prt}, Agt_k, g_k, p_{qk}, a_m)) \\
& \quad \wedge (\neg \exists AGT_m AGT_j \subset AGT_m dep_{and}(agt_{prt}, AGT_m, g_k, p_{qk}))
\end{aligned}
\tag{2.11}
$$

In short, the Equation 2.11 shows there is only one set of agents Agt_j that can perform the actions and required by the agent agt_{prt}. Once again, the recursive definition in the Equation 2.11 shows the set of agents Agt_j is a maximal set. Figure 2.7 illustrates the AND dependence relationship dep_{and}.

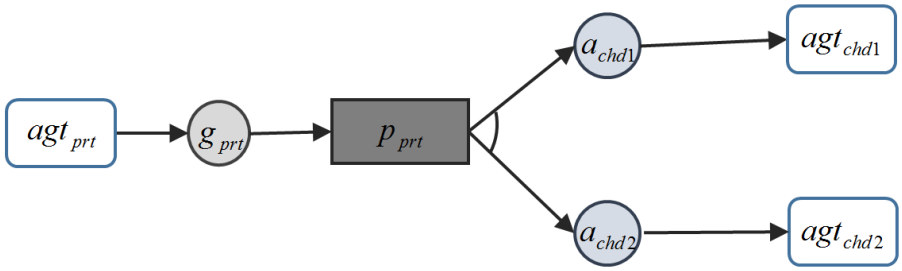


Figure 2.7 The AND-dependence relationship

The AND-Dependence from the Figure 2.7 shows the agent agt_{prt} has no option but to choose both agents $\{ agt_{chd1}, agt_{chd2} \}$ for the required a_{chd1} and a_{chd2} to achieve the goal g_{prt} under the plan P_{prt}.

Chapter 3 Background

In this chapter, existing literature and key information related to our proposed solutions are presented. In Section 3.1, we will discuss the dependence relationships between agents based on social reasoning mechanism (SRM). This is the only mechanism currently available to identify agents' dependence relationships. In Section 3.2, we will go into detail of the coalition's background while Section 3.3 summarizes this chapter.

3.1 RELATIONSHIPS BETWEEN AGENTS

The *social reasoning mechanism* (SRM) (J. S. Sichman et al., 1998) was introduced to help agents to identify their relationships during collaboration. It aims to mitigate the communication overhead while fulfilling each agent's social need (Castelfranchi, 1998) by choosing the ideal partner. It is based on *dependence theory* where agents have to depend on each other due to their heterogeneous characteristic. As suggested by the heterogeneous characteristic, agents are not autonomous of their goal (not *S-autonomous*) since they do not have the required actions and resources to achieve the goals.

3.1.1 Dependence Theory

Dependence theory (Conte & Sichman, 2002) is crucial in investigating the fundamental aspects of interdependence relationships between agents during their interaction. A heterogeneous system consists of agents that are composed of different capabilities and have different needs. An agent will depend on other agents in term of actions and resources to achieve its goals due to limited autonomy (Guido, Torre, &

Villata, 2009; J. S. Sichman & Conte, 2002). The limited autonomy implies an agent is not capable of performing the required action or providing the resource itself to achieve the goal. According to Grossi and Turrini. (2010), there is no well-defined standardized dependence theory yet in MAS. However, our proposed mechanism follows the dependence theory as following:

Definition 3.1: Dependence theory is represented by $((agt_i \neq S\_aut) \vee (agt_i \neq A\_aut)) \wedge ((\neg\exists a_i \in p_{agt1}) \vee (\neg\exists r_i \in p_{agt1}))$ where an agent is not autonomous of its capabilities and required to depend on others.

It means an agent is not able to achieve its goal without the required capabilities. After denoting dependence theory, we can categorize types of dependence relationship occurring in between agents. There are two fundamental dependence types as shown in the following Definition 3.1.1 and Definition 3.1.2:

`

Definition 3.1.1: Action dependence shows an agent agt_i depends on agent agt_j for the specific action, a_k.

Definition 3.1.2: Resource dependence shows an agent agt_i depends on agent agt_j for the specific resource, r_k.

Using the two dependence types from above, an agent can depend on others agent for action and resource. To further understand how dependence relationships between agents work in a society, the following Example 3.1 describes relationships between agents.

Example 3.1*:* The agent agt_1 needs to translate a paragraph but it does have the specific knowledge to translate. However, agent agt_2 has the translation skills with the dictionary and is willing to help agent agt_1. Through the definition of action-dependence (a-depends), the agent agt_1 will look for agent agt_2 and will request the agent agt_2 to translate. If the agent agt_2 accepts agent agt_1's offer, agent agt_2's current tasks will be interrupted. On the other hand, the Resource-dependence (R-depends) scenario suggests agent agt_2 will simply lend a dictionary to agent agt_1. The agent agt_2 will not be interrupted by agent agt_1 as the agent agt_1 can perform the translation itself by referring to the dictionary.

Based on the case study above, a-depends is given a higher priority than r-depends in the dependence network and graph. The following Figure 3.1 shows an example of a-depend and r-depend by utilizing the dependence graph:

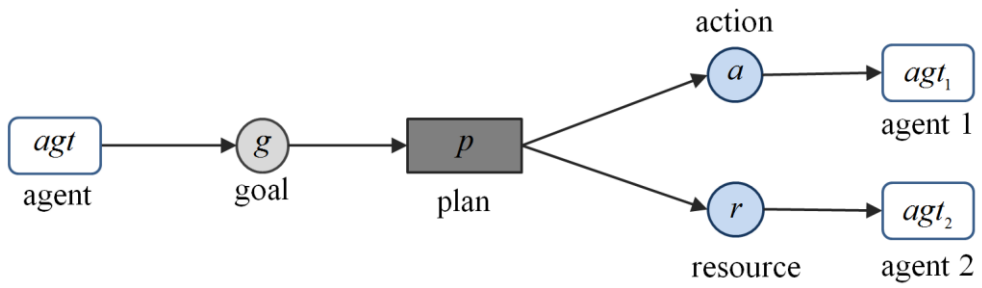


Figure 3.1 The a-depend and r-depend

3.1.2 Dependence Network

Dependence relationship table is the formal tools to identify an agent's dependence relationship. The dependence relationship between agents are represented by the dependence network to visualise a simpler interdependence relationships. First, the

dependence relationship table will be constructed to view individual agent's relationship. Later on, the extending line will be used to visualise the dependence relationship of an agent and it is called depdnence network. Consider the following scenario in Table 3-1 and will be used in the next subsection.

Table 3-1 The dependence relationship table

agt_j	$G(agt_j)$	$A(agt_j)$	$R(agt_j)$	$P(agt_j)$
agt_1	g_{11} g_{12}	a_1	-	P_{g11} P_{g12}
agt_2	g_2	a_2	-	P_{g2}
agt_3	-	a_3	-	-
agt_4	g_4	a_4 a_5	-	-
agt_5	g_5	a_6	-	P_{g3}

Using Table 3-1, we can form the dependence network using horizontal extending line. Figure 3.2 shows the dependence network based on the Table 3-1:

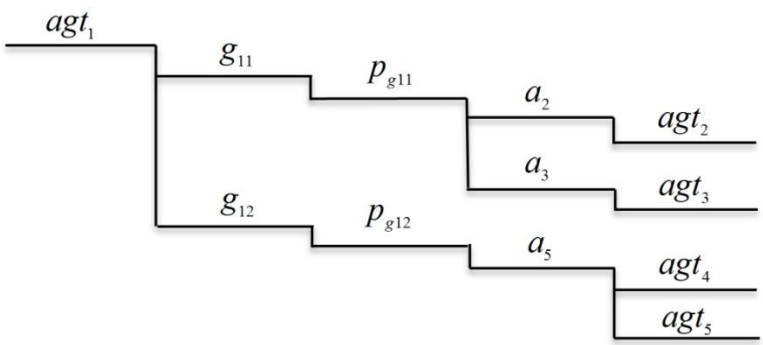


Figure 3.2 The dependence network based on Table 3-1

3.1.3 Dependence Graph

The dependence network suffers from the limitation such as complex extending line if the numbers of the relationships grow. Sichman and Conte (Conte & Sichman, 2002; J. S. Sichman & Conte, 2002) have further improvised the dependence network by introducing the dependence graph. It is based on *Graph theory* (Bondy & Murty, 1976) where every agent is represented as nodes along with its external descriptions. Figure 3.3 shows the dependence graph for agent agt_1 (note: The other agents' dependence relationships have been omitted for easier representation).

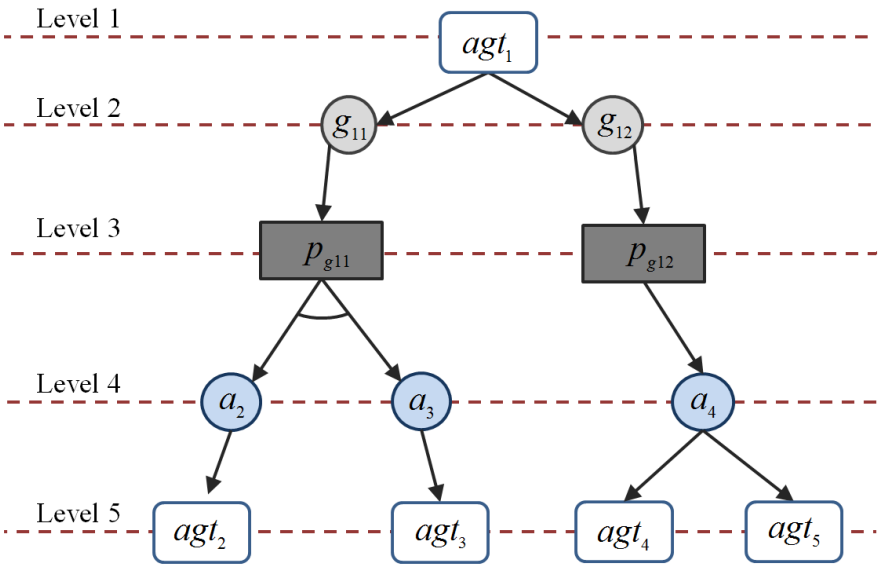


Figure 3.3 The dependence relationship graph for agent agt_1

The Figure 3.3 shows a graph of dependence network which consists of five different agents with two goals of g_{11} and g_{12}. The starting and ending nodes are the depender and dependee respectively. Both nodes are linked by the goal's node which acts as the main objective of forming the dependence relationship. Plan and action nodes are allocated under the goal node to express how agents are going to achieve their goals. Normally, the dependence graph has a depth of five levels as shown in Figure 3.3. However the dependence graph will be more complicated if the number

of agents involved in the dependence relationship increases. Hence, a reduction graph has been derived to simplify the dependence graph. Figure 3.4 shows a reduced version of a dependence graph by neglecting plans and goals nodes in the Figure 3.3:

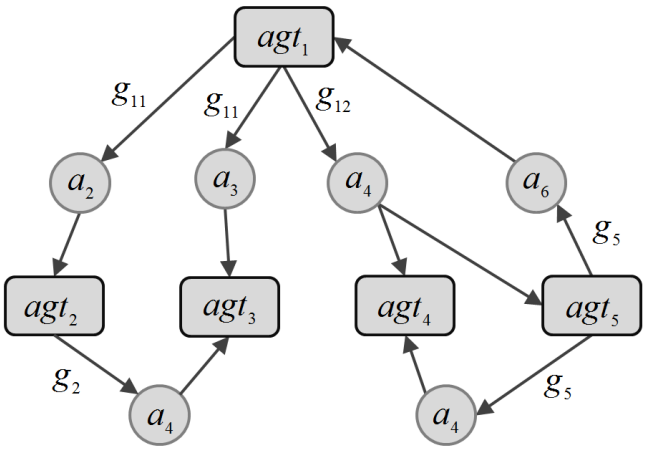


Figure 3.4 The reduced dependence graph based on Figure 3.3

3.1.4 Dependence Relationships

There are three variations of the dependence relationships for representing the collaboration of agents. According to Sichman et al. (J. Sichman & Demazeau, 1995; J. S. Sichman et al., 1998; J. S. Sichman & Demazeau, 2001), the dependence relationships are classified into AND, OR and SINGLETON relationships. The following Table 3-2 shows a general explanation of the dependence relationships:

Table 3-2 Variations of the dependence relationships

Dependence relationships	***Explanation***
OR-dependence, dep_{or}	This relationship indicates that an agent has the option of choosing between certain agents to cooperate for achieving its goals.

(Table 3-2 Continued)

AND-dependence, dep_{and}	This relationship indicates that an agent would have to take into account every agent involved in the dependence relationship to achieve the goal.
SINGLETON-dependence, dep_{1to1}	The singleton relationship indicates that the agent do not have other option of choosing other agents to cooperate to achieve its goals. It only has one partner to invite to form the dependence relationship.

The OR-Dependence is one of dependence relationships which is given a higher priority. The agent engaged in dep_{or} relationship will just choose a convenient partner to form collaboration. On the other hand, the AND-Dependence will have to include all the involved agents to form collaboration. It has been shown by An, Shen, Miao, and Cheng (2007) that the AND-Dependence has a higher complexity to form the cooperation. However, capacity of each agent is not accounted because an agent can join more than one coalition in certain scenarios (this is also known as coalition overlapping).

By understanding dependence relationships, it helps a cooperative agent to determine how a common goal can be achieved through the selection of the ideal partner. However, the direct dependence is not able to express the overall view of relations at an organization level due to agents' limited knowledge. Hence, transitive dependence relationship (T-Dep) (An, Miao, Tang, Li, & Cheng, 2005) has been developed to address the indirect dependence between agents. The main component of T-Dep is dependence chain (Dpc) that describes the transitive property of a relationship. The Dpc is composed of head and tail agents and is denoted as following:

$$Tdep(Head(Dpc), Tail(Dpc), Dpc)$$

Figure 3.5 shows the example of a T-Dep relationships between agent agt_1, agt_2 and agt_3 with the common goal g_1.

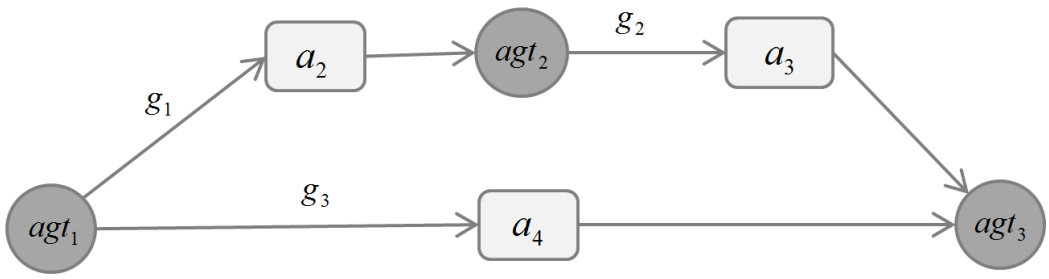


Figure 3.5 Example of the T-Dep relationships

T-Dep relationships in Figure 3.5 and can be denoted as $dep = TDep(\ agt_1, agt_3,\ depchain\ )$ where the Dpc is:

$$depchain = agt_1 \xrightarrow{g1,a1} agt_2 \xrightarrow{g2,a2} agt_3$$

An et al. (2007) has applied transitive dependence relationship to solve the problem between a series of furniture, cabinet and wood factories. Despite the proposed transitive dependence relationship's incomplete payment configuration, agents are able to form coalition using the greedy solution approach. It was implemented using a virtual organization (Petrie & Bussler, 2003) that allows the agents to work on a temporary basis (Travica, 1997).

3.1.5 Discussion

There are other extensions proposed to extend the SRM features such as dynamic element, indirect relationship and social degree of the dependence relationship. The following Table 3-3 shows a compilation of the existing extensions:

Table 3-3 List of extensions associated with SRM

Type of Extensions	***Features***	***Drawbacks***
Transitive dependence (An, Miao, & Cheng, 2005; An, Miao, Tang, et al., 2005; An et al., 2007)	Indirect dependence of the agents. The dependence chain has been formulated for identifying bigger chain of network in the society.	The dependence relationships are limited to the current state. It is also unable to address the dynamic element of the relationship.
Dynamic dependence 1 (Caire, Villata, Boella, & van der Torre, 2008)	It is consists of three tuples: $< A, G, dyndep >$. The goals and powers of the agent are conditional and can be changed overtime.	Did not include the role of agents. Did not state how to account for individual agent's availability overtime.
Dynamic dependence 2 (Guido, der, & Serena, 2008)	It also consists of three tuples: $< A, G, dyndep >$. However, it combines two abstract views - (1) power view and (2) mind view.	Only considers agents goals in order to form coalitions. Role of each agent might vary.
Temporal Dependence Network (Cair & Torre, 2009)	A tuple of $< A, G, T, dep >$ where T is the set of natural numbers for forming relationship. Provides a sequence diagram and goal diagram for modelling the problem.	It is a conceptual model and does not have actual implementation.

(Table 3-3 Continued)

Quantifying degrees of dependence (Costa & Dimuro, 2007)	Using Dependence Situation (DS) Graphs to measure the dependence degree of relations.	Did not consider the qualities of the relationships and emphasizes on the quantity of the relationship.
Coalition formation (An, Miao, Tang, et al., 2005; An et al., 2007)	Provides algorithms for identifying (1) without and-action dependence and (2) with and-action dependence. They also provide the complexity analysis of the algorithm.	The SRM model used does not consider communication between agents.

Dependence graph has been a success in investigating the dependence relationship between agents. However, it suffers from a few issues when applied in a dynamic environment such as OMAS (Guessoum, Ziane, & Faci, 2004). The potential issues encountered when applying SRM are addressed in the following subsections:

3.1.5.1 Dynamic Properties

In dependence graph, there is assumption of the auto knowledge principle states that an agent has a complete representation of themselves and others. This is not applicable in the real world scenario as there are many uncertain elements require an agent to learn about. Mainly because in most scenario, an agent has incomplete information about others and themselves. Methods such as “Learn by being told” (Nwana, 1990) has been proposed to address learning in dynamic environment. Besides, the assumptions of sincerity principle have omitted the possibility of malicious agents which provide inconsistent and incorrect information.

Consider an agent agt_j depends on agent agt_i to achieve its goals but agent agt_i leaves the society in the middle of committing tasks. In the current state, the agent cannot achieve its goal because the resources or actions required are missing from the society. Hence, fault handling techniques have been developed to combat against the uncertainties during collaboration.

One of the dynamic issue solutions includes technique of replication (Guessoum et al., 2004) and it is common for fault handling built inside an agent. However, it is not practical when the number of agents becomes large. The replication and backup mechanism is impractical in a big agents society as the memory consumption will grow exponentially. Thus, a more efficient methodology is required to mitigate the problem of a dynamic element.

One possible improvement is to implement selective replication of agents with unique actions or resources. Through this mechanism, agents are guaranteed to have a minimum of two alternate sources for the required resources or actions. In order to realize this mechanism, we assign an agent with special role to traverse the MAS society to search for agents with unique resources or actions.

3.1.5.2 Agents' Economic State

The economic status of an agent is believe to have impact on its rate of cooperation. Unfair allocation of payout distribution causes the cooperation between agents to decrease. Dynamic dependence graph only addresses the current state of the agents and assume agents help each other to achieve goals by contributing actions and resources for some profit. From economics perspective, agents are self-interested

entities and will not help others if there is no profit.

Some authors have included cooperative game theory in order to generate optimal cooperation between agents to improve overall social welfare. The Shapley value (Roth, 1988; Lloyd S. Shapley, 1953) is the value that is used to calculate the weighted average utility of the agent that contributes to all possible combination of cooperation. Shapley value emphasizes on the fairness payment of all involved agents. A simple mechanism for sub-additive tasks has been introduced by Zlotkin and Rosebschein (1994). However, the computation of Shapley value in term of space and time complexity is high.

Alternately, the kernel of a cooperative n-person game (Davis & Maschler, 1963; Morton & Michael, 1967) has proposed that configurations for cooperation are stable if an equilibrium exists between pairs of agents in the cooperation. In addition, Hossein, Taghi, Nicole, and Hamid (2010) has introduced nucleolus and lexicographic into cooperative game and was able to solve linear programming problems. However, the calculation of this mechanism is not suitable for real world problem as there are exponentially high uncertain elements to be taken into calculation.

There exists a concept of choosing ideal agent to collaborate in a dependence relationship that based on accumulating weight of individual agents such as finding least cost root of vertex (An et al., 2007). It chooses agents to form coalition based on the least cost selection by considering AND and OR dependence. However, it does not guarantee the equilibrium of the society. Besides, there are also some

models related to social exchange (M. Rodrigues, Costa, & Bordini, 2002; M. R. Rodrigues & Costa, 2003; Schillo, BУMrckert, Fischer, & Klusch, 2001) where the exchanging mechanism in the society is implemented to exchange information of payoff distribution for verification purposes.

Pistolesi and Modesti (2001) have proposed a concept of budget to control the parameter of the specific resources usage during cooperation between agents. This is one of the interesting models that might actually function in a large scale MAS society. However, it is not able to address the society of agents in large scale.

3.1.5.3 Agents' Credibility

The credibility of an agent infers the trustworthiness of a particular agent in the MAS society. The credibility of an agent will have impact on choosing which agents to depend on. In normal circumstances, agents with lower credibility will be given the lower priority during tasks allocation. On the other hand, malicious agents will attempt to gain the highest profit by cheating others or spreading fake information. This scenario will "ruin" the MAS society as the malicious agents can manipulate its partner.

To address the credibility issues, a de-commitment value of an agent (Komenda, Vokrinek, & Pechoucek, 2011; Vokrinek, Komenda, & Pechoucek, 2009) serves as a parameter to control the dependence between agents based on their commitment. If an agent leaves the MAS society during the tasks execution, it will be given a de-commitment penalty depending on the "excuses". Through these sequences, an agents' trustworthiness is being moderated according to the "excuses".

The credibility of agent is represented through its belief state with the deduction of reasoning module. The complete representation of the agents in MAS is a challenging task because the incomplete information can be abused for committing malicious attack. One possible solution is to use communication channel to gain more knowledge about the environment or percept near-complete information about the environment to make a better decision. Through these actions, agents might suffer intense communication flow with the limited resources which require further investigations and enhancements to resolve it.

3.1.5.4 Agents' Working State

The working state of an agent refers to its current conditions and states during the occurance of dependence relationships. The agents' status can be, for example; (a) engaged in performing certain actions, (b) idle or (c) available for achieving other agents' goals. The dependence graph does not represent the status of an agent when a dependence relationship occurs. It only addresses the current agent's relationship based on the target of achieving goals.

The status of an agent is related to its role in the dependence relationships. Examples of the roles' strategy proposed are substantialist, utilitarian and misers (de Lima do Rego Monteiro & Sichman, 2006). The agent's roles decide the behaviour of the agents such as optimized for goal, searching for ideal partnership to form coalition and so on. It should be one of the elements considered during the design of MAS architecutre.

A possible solution for the tasks delegation issues is to create a resource management agent to plan, execute and monitor the tasks. It is to help agents manage their workload. Current well-known approaches for tasks allocation and resource handling include market based allocation (Bredin et al., 2001; Chavez, Moukas, & Maes, 1997) and game theories (Bredin et al., 2000; Feldman & Tennenholtz, 2010).

3.2 MANAGEMENT AND PROCESS OF COALITION

According to Horling and Lesser (2004), the coalition is a goal-directed organizational framework where the agents cooperate to solve the problem within a short-lived interval. When there is no common goal or the coalition is no longer needed, the coalition will simply dissolve. However, there are long-term CF cases (Blankenburg, Dash, Ramchurn, Klusch, & Jennings, 2005) that agents work together utilizing a trust mechanism to achieve longevity worthiness.

An agent can join more than two coalitions in parallel and this phenomenon is represented as overlapping coalition (Shehory & Kraus, 1998). It allows an agent to join more than one coalition at a time. Also, a nested coalition is also possible if there is no explicit hierarchical characteristic for CF. Figure 3.6 shows an example of the CF with one overlapping agent.

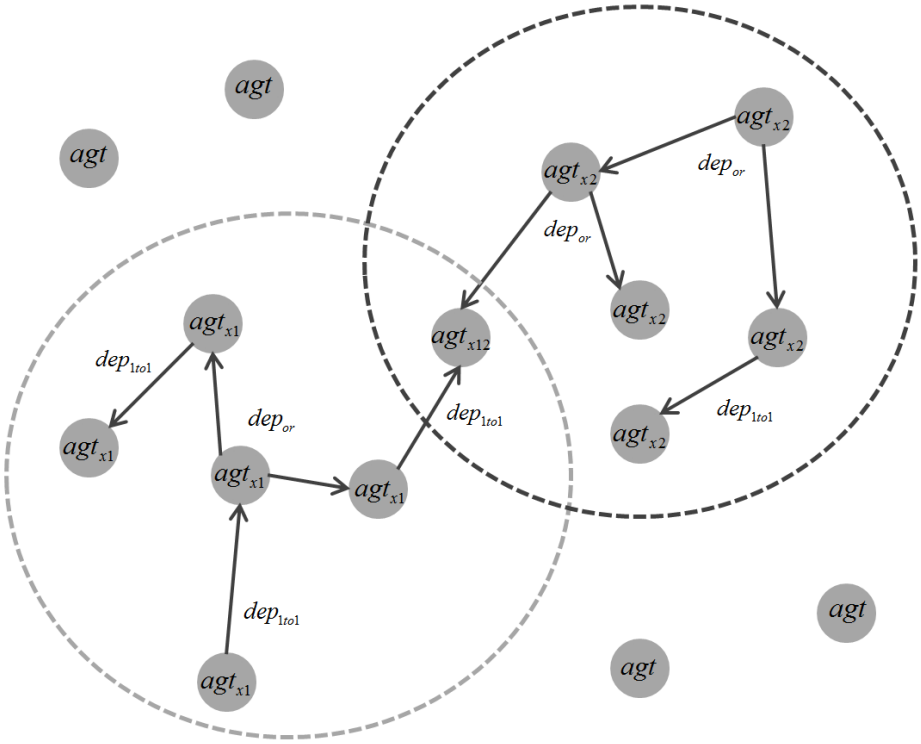


Figure 3.6 Example of a CF with one overlapping agent

3.2.1 Coalition Management

There are two common types of coalition managements which are (1) microscopic and (2) macroscopic coalition. The main difference between these two coalition management is the presence of the coalition leader in the organization affair.

3.2.1.1 Macroscopic Perspective

The macroscopic coalition is defined as a leaderless organization for a group of agents to form cooperation. Due to its flat architecture, the communication for deciding tasks planning and delegations between agents are computationally expensive. This is because the coordination of tasks requires every agent to be involved. Breban and Vassileva (2002) has shown the public voting inside a macroscopic coalition is both time and resource consuming. However, the macroscopic management implements the distributed decision making approach

among coalition members and it is better at fault handling than the centralised paradigm, microscopic coalition.

3.2.1.2 Microscopic Perspective

The microscopic coalition has a coalition representative to manage the organization affairs. The coalition representative (coalition leader) serves as a central control unit for planning, scheduling and allocation of tasks to the coalition members. The presence of the coalition leader lead to a better communication and management efficiency among coalition members. The decision making of a microscopic coalition varies depending on the leader's authority. Two microscopic based coalition sub-management approaches are derived as following:

- Democracy based coalition

The democracy based coalition implies a leader does not possess full authority in decision making on behalf of the coalition. Every coalition member has to contribute their decision through the voting session proposed by the agent agt_{lead}. Normally, a voting session will occur when there is major decision that requires attention of all coalition member. However, a leader agent agt_{lead} is a veto agent during the voting session and it holds the priveledge for cancel the voting session if required.

- Authoritarian based coalition

The authoritarian based coalition implies a leader has the full authority for making decision on behalf of the coalition. It is also known as dictatorships where a leader agent agt_{lead} has full power over every coalition affair. There will be no voting session held among the coalition members as the leader agent agt_{lead} decides for the

coalition. The communication between agents is deemed to be minimized but this approach does not consider the minority side of the coalition members during decision making.

3.2.2 Formation Process

The formation of a coalition can be divided into the following three processes:

- Coalition value calculation
- Coalition structure generation
- Payoff distribution calculation

3.2.2.1 Coalition Value Calculation

The CVC is the allocation of numerical numbers assigned for every coalition member which produces a *coalition value*. It assigns each coalition member a number for distributing the payoff configuration before calculating the overall cost and profit. The calculation of the coalition value relies on investigation of problems before assigning a value to each coalition member. This is normally performed by agents who proposed the CF and the computational complexity varies depending on the depth of dependence relationship's investigation.

Based on uptodate literature review, there are only two existing methodologies that perform the calculation of coalition value. They are Shehory and Kraus (1998)'s and Rahwan and Jennings (2007)'s algorithm. Shehory and Kraus's algorithm requires intense communication between agents and it is exponentially hard to compute as the number of agents increase. Agents are required to communicate several times to send a list of the related information on coalition value to each other. This is because all

the information is locally encapsulated inside an agent's knowledge and normally not accessible by other agents. The Rahwan and Jennings's algorithm uses a decentralized approach to perform CVC which further reduces the communication between agents. Their results show a significant improvement over the existing Shehory and Kraus's algorithm in term of memory usage, time complexity and communication overhead. The Table 3-4 shows the difference between the two CVC algorithms:

Table 3-4 The comparison between Shehory & Kraus' and Rahwan & Jennings' algorithms

	Shehory and Kraus (1998)'s algorithm	***Rahwan and Jennings (2007)'s algorithm***
Communication overhead	Exponential (Centralized calculation)	Minimal (Decentralized calculation)
Memory usage	Information is passed around with the member and exponential increase communication with agents' number	Minimal memory usage but requires enough memory space to maintain distribution calculation.
Management paradigm	Centralized	Decentralized
Quality of the value generated	Does not guarantee the fairness of the value generated.	There is less redundancy in the calculation of each agent and equality of agents is guaranteed.

Although the CVC algorithm proposed by Rahwan and Jenning is decentralised, the complexity of assigning a value to each coalition member is still a NP hard problem. The reason behind the complex calculation is the number of elements considered during the computation of the coalition value. It has not received much attention

from researchers as each system has different variation of elements for consideration. According to Metcalfe's law (Shapiro & Varian, 1999), basic communication between agents in a coalition tend to be proportional to their number which is n^2. Despite that, some researchers (Ross, 2003) have shown Metcalfe's law is not accurate in a real world scenario and varies depend on the communication protocols.

3.2.2.2 Coalition Structure Generation

The CSG investigates the problem of partitioning agents' set into singular or multiple disjoint coalitions. The end result of the partition process is called generating coalition structures. It provides agents a domain for cooperation but does not cover the coordination and communication process between coalitions. Given a set of agents $Z = \{agt_1, agt_2, agt_3\}$, the possible coalition structure are:

$$\{\{agt_1\},\{agt_2\},\{agt_3\}\},\{\{agt_1\},\{agt_2,agt_3\}\},\{\{agt_2\},\{agt_1,agt_3\}\},\{\{agt_3\},\{agt_1,agt_2\}\},$$
$$\{\{agt_1,agt_2,agt_3\}\}$$

It is a common practice for the *characteristic function games* (CFG) to be included in CSG (Kahan & Rapoport, 1984; Ketchpel, 1994; Osborne & Rubinstein, 1994; Sandholm et al., 1999; Shehory & Kraus, 1996). The value of each sub-coalition has been assigned a structure through the characteristic function. These values represent the quality of the coalition generated. The main goal of performing coalition partition is to search for an optimal coalition using low computational complexity. In this thesis, these values are generated by the combination value of goal, priority of tasks, budget and trust ratio.

Sandholm et al. (1999) has shown the partitioning of coalitions is a NP-Complete

hard problem. They have developed their own algorithm and prove the lower bound of searching an optimal coalition structure by comparing other existing algorithms (Ketchpel, 1994; Shehory & Kraus, 1996). Furthermore, the detail properties of the CSG problems are listed in the following:

- Anytime versus design-to-time algorithms

Anytime algorithms suggest the attempt of searching an optimal coalition within a given time based on the initial value inputted. Once a feasible result has been obtained, it will continue to refine its result until the termination of algorithms. On the other hand, Design-to-time algorithms (Garvey & Lesser, 1993; Zilberstein & Russell, 1996) use a probabilistic model to establish searching of the optimal coalition with restriction of boundaries or constraints. From an efficient algorithm's perspective, design-to-time algorithm is more ideal in searching optimal coalition given the result will be polished overtime to generate a more optimal coalition structure.

- Off-line versus On-line search control policies

The term offline and online represents amount of information that flows between agents. The offline control policy shows agents depends on itself for the knowledge gaining when generating coalition structure. It does not possess global knowledge during the searching process. The online searching control policy is an opposite approach where the agents' information is shared globally. In term of searching efficiency, the online control policy provides a promising result as agent is able to evaluate at the organization level. However, this might burden agent in term of communication intensity as huge amount of information need to process before

reaching a global state.

- Observing coalition structure values versus observing individual payoff vector

If an agent has the intention of monitoring the values of the coalition structure instead of individual payoff vector, these analogies are similar to search space problems with weight property. These includes weighted set packing (Hochbaum, 1997), knapsack problems (An et al., 2007), weighted maximum clique (Bomze, Budinich, Pardalos, & Pelillo, 1999) and determination of winner in combinatorial auctions (Sandholm, 2002). The observation of the individual value emphasize on agent's welfare than the overall profit.

- Centralized versus distributed

In a microscopic coalition, a centralized approach is implemented to ensure the coalition leader has control over the members. This management approach suffers from heavy workload as the leader is required to handle CSG computation by itself. On the other hand, the macroscopic coalition advocates a distributed approach to compute the CSG. This approach requires agent to have up front knowledge of the coalition values where communication is important in updating all agents' latest knowledge.

- Degree of generality

The coalition structure generated does not guarantee the agents' interoperability across different systems or platforms. This is one of CSG criteria where a coalition structure generated is able to use in other systems. A variety of considerations have

been implemented into the design of algorithm. It can be effective in increasing the generality of coalition structure generated as proposed in the Ketchpel (1994)'s and Shehory and Kraus (1996)'s methods. However, both methods' lower bounds are not rectified and hold no promises on generating the optimal CSG.

Based on the five properties of CSG algorithm discussed earlier, the classification of algorithms are described in the next subsection. The quality of the result generated relies on the depth of investigation of the coalition structure. In a real world scenario, optimal solution does not necessarily denote optimal computational complexity in most situation. It might compromise certain performance measurements in order to achieve optimality in any criteria. However, a feasible result is also known as a "good" result because it ensures the overall profit of individual agent. Generally, the CSG's end result can be classified into three patterns as following:

3.2.2.2.1 Optimal Solution with High Computational Complexity Algorithm

This approach shows an optimal CSG solution is generated but with a higher computational complexity. One might argue that an optimal solution is preferred in most situations but it is essential for system when the end result accuracy is critical. The ultimate goal is to obtain the optimal result regardless of the computational time. The number of agents have become the main factor as it can exponentially increase the computational time. Dynamic programming is a common technique (Leiserson, Rivest, Stein, & Cormen, 2001) used to obtain optimal solution for overlapping coalition problem and optimal substructure. It has been applied to solve the complete set partitioning problem by Yun Yeh (1986) and Rothkopf, Pekeč, and Harstad (1998). The major limitation of these algorithms is inability to generate a solution within a short interval and is only able to address the small number of agents in a

coalition efficiently.

3.2.2.2.2 Non Optimal Solution with Fast Forming Algorithm

As suggested by the name itself, the algorithms aim to return a feasible result as soon as possible but do not provide a guarantee on the computational and time complexity. Heuristic models such as genetic algorithms (Sen & Dutta, 2000; Yang & Luo, 2007), neural network (Abdallah & Lesser, 2004) and greedy approach (Rahwan, Ramchurn, Jennings, & Giovannucci, 2009; Shehory & Kraus, 1998) are able to generate a feasible solution for CSG with some drawbacks. The algorithm advocates the iterative approach in stages and each stage consists of attempt to generate the highest coalition value through partitioning of coalition structure.

3.2.2.2.3 Solution within the Bounds with Anytime Algorithms

In a real world scenario, obtaining an ideal solution is inevitable as the searching space can be exponentially complicated. In order to deal with these uncertain factors, various approaches (An et al., 2007; Sandholm et al., 1999) have been proposed to generate coalition structure given a boundary to the computational time. This ensures the worst case of the searching and possible refinement of result if there is allowance on time. There are not many anytime algorithms currently available as the one cannot guarantee the lower bound to generate a feasible solution. Sandholm et al. (1999) and Dang and Jennings (2004) are among the few that have proposed the CSG with anytime algorithms. Both algorithms use similar approaches but Dang and Jenning's algorithm performs better in a smaller domain while Sandholm's algorithm has a smaller bound if the partitioning of the coalition structure become complicated due to increasing agents' number.

3.2.2.3 Payoff Distribution

Currently, there are three well-known payoff distribution methodologies proposed which are Shapley value (Lloyd S. Shapley, 1953), Core (Klusch & Gerber, 2002) and Kernel (Morton & Michael, 1967). The Shapley value is the value used to calculate the average utility's weight of the particular agent for every possible distribution among the members in the coalition. It emphasizes the fairness payment for each coalition member. However, the computation of Shapley value in terms of space and time complexity is high and not practical. This is because it considers all possible subsets of CF in order to generate an optimal payoff distribution. The kernel of the cooperative n-person game suggests that the coalitional configurations are stable when equilibrium exists between pairs of individual agents in the same coalition. The kernel is an essential subset of bargaining set where it assigns payoff to various coalition structures without labelling each player's payoff configurations. On the other hand, Hossein et al. (2010) have introduced nucleolus and lexicographic kernel into the cooperative game and was able to solve natural linear-programming formulation. However, the algorithm proposed is not suitable for tackling the real world problem as there are exponentially element of vectors to consider when calculating the payoff for each agent. The core of the game is the data structure in CF that holds unique payments for each agent in a coalition. Despite being core-stable is able to maximize the sum of coalition values of a coalition structure, searching for an optimal coalition is high computational complexity and hard to implement. Philip, Eyal, and Myrna (2012) have introduced the partner core of a game with side payments to address the non-asymmetric dependencies between two players. It has greatly reduced the number of agents required for calculating the core which provides the payoff distribution calculation with lesser information is needed. It is

not required to communicate with previous partner agent for extra information during calculation. Some researchers (Chalkiadakis & Boutilier, 2004; Georgios & Craig, 2012) have removed the unrealistic assumptions such as uncertainty during the payoff value calculation and proposed a model using the Bayesian reinforcement learning module. Through this, agents are able to calculate the payoff distribution regarding the dynamic property of its environment. The following Table 3-5 shows the comparison of each payoff distribution methodologies used for CF:

Table 3-5 Comparison of the payoff distribution

Payout Method	***Finding and Features***	***Drawbacks***
Shapley Value	Emphasize on the fairness of agents' payoff distribution.	Hard to compute as it involves every possible set of coalition (permutation).
Kernel	Does not depend on the labelling of agents. Easier to compute compared to the bargaining set of the games.	Exponentially hard to compute when elements of a vectors are increasing.
Core	Does not require unique payoff distribution for each agent. Based on the stability concept.	Searching an optimal coalition structure is exponentially hard. The core-stable configuration might be empty which is exponentially hard to compute.

3.2.3 Discussion

The formation of dynamic organization such as coalition has brought the current MAS a step nearer to OMAS. However, to reduce the gap between MAS and OMAS, coalition paradigm must be able to address the dynamicity of OMAS environment. The following issues arised when we are moving forward from coalition to OMAS:

3.2.3.1 Mutual Trusts

Trust between agents is the key for agents to cooperate while maximizing the mutual profit. According to Jonker and Treur (1999), trust can be defined as the attitude of an agent with respect to other agents' capabilities or dependability corresponding to the flow of events. The long-term coalition has been studied by Blankenburg et al. (2005) and Nathan and Luck (2003) which is against the traditional design protocol of the coalition. Blankenburg et al. has modified the traditional kernel-based CF by introducing a mechanism that chooses most the reliable partner. A trust model is implemented to ensure a reliable agents are agents from the previous cooperation that has successfully secured a profitable collaboration. Nathan and Luck has introduced the concept of motivation and implemented it into the reasoning module. The reasoning module helps an agent to make local decision with the mindset of obtaining mutual benefits. Through this, agents are able to form the coalition and try to help each other with the basic expectation of profit is guaranteed. In the remainder of this thesis, agents in the coalition are assumed to be cooperative agents. It implies agents are willing to cooperate with each other to improve the social welfare.

3.2.3.2 Completeness of Coalition Formation's Activities

Most of the research works have been focusing on the CSG and payoff distribution. CVC has not been receiving attention from researchers because different systems have unique criteria for calculating coalition value. Each formation process is important as it affects the welfare of each agent during cooperation in fulfilling shared goals. Major well-known research works (Rahwan et al., 2009; Sandholm et al., 1999) did not consider the initial stage of formation process such as CFP (i.e. CNP (Smith, 1980)).

Moving on, the notation of relationship between agents during cooperation is also not formarly addressed. The dependence relationship between agents in the coalition has been a missing puzzle piece until the Gaspar and Morgado (2000) implemented the SRM into the CF. Through the implementation, an agent is able to perform social reasoning to identify more efficient partner to cooperate. However, the proposed method is not complete as there is no methods of implementation into a real world scenario application. We strongly believe that the understanding of the social reasoning can help agents further reduce the intermediary during cooperation. This can further reduce the communication overhead generated during communication. Hence, from an organizational perspective, individual reduction of communication overhead will lead to overall reduction of unnecessary communication in the coalition. This has been proposed by Boella, Sauro, and Torre (2004) as one of the agents' perspective which includes mind view, dependence view, power view and coalition view. On the other hand, the information level of dependence has been presented by Fan, Wang, Sun, Sun, and Yen (2005) in conjunction of addressing difference level of interdependence. This helps agents to further gain knowledge about the organizational level view when cooperating with others.

3.3 SUMMARY

In this chapter, we have discussed the dependence relationship between agents in several visualization forms such as dependence network and graph. The limitation of the SRM has been outlined such as the inability to address the dynamic element especially with the CF. Although some researchers (Gaspar & Morgado, 2000) have successfully merged the SRM into coalition, the coalition still suffers from certain

limitations. The existing SRM is incomplete to address different management approaches such as macroscopic and microscopic. With this in mind, we have developed the join coalition mechanism (JCM) to address different management approaches along with CVC and verification of agent's dependence relationship.

- The development of budget mechanism for transitive dependence based CF to generate coalition value generated is done locally. It ensures the quality of the coalition value generated and minimizes communication overhead (as shown in the Chapter 4). However, the verification of the dependence relationship is discussed in next Chapter.

- The implementation of self-verification algorithm for agents' relationship (Shown in Chapter 5) during cooperation in a coalition. This ensures all the agents in coalition are contributing to the overall profit and ensuring the stability of the coalition.

- Once the verification of the coalition dependence relationship have been confirmed, we will focus on increasing global utilitism of the coalition. To achieve that goal, we have introduced mechanisms (shown in Chapter 6 and Chapter 7) that allows an external agent to join the coalition using interruption. Two types of coalition managements have been addressed in the proposed mechanism which are macroscopic and democracy based microscopic coalition.

Chapter 4 Transitive Dependence based Coalition Formation

4.1 INTRODUCTION

With the foundation of the dependence theory, An et al. (An, Miao, & Cheng, 2005; An, Miao, Shen, Miao, & Cheng, 2005; An, Miao, Tang, et al., 2005; An et al., 2007) has proposed the T-Dep based CF. T-Dep based CF aims to address the indirect relations between cooperative agents in a coalition. However, the payment configuration of each agent engaged in T-Dep based coalition is not addressed. The proposed mechanism only involves the CSG where the remaining of coalition activities are unclear and incomplete. This shows the incompleteness of transitive dependence based CF as it does not ensure individual welfare of coalition members. The coalition can suffer loss of profit overtime because there is no mechanism to check the cost and profit of organization. From a long term perspective, agents will be demotivated as there is no guarantee on the profit earned through the coalition.

This chapter deals with the CSG problem based on the T-Dep based CF. It consists of cooperative agents that utilizes the indirect relations to achieve their goals. We have developed the T-Dep based budget mechanism (T-DepBM) to address the CSG and CVC problem. It has been implemented into T-Dep experiment systems (T-DepExp system) (Lau, Singh, & Tan, 2012) which was originally developed to study the agent's relationships. A heuristic and greedy approach has been implemented into T-DepBM which uses budget as the constraint for formation of a coalition. This chapter is organized as follows: in the first section, a literature review of T-Dep based CF is

presented followed by the description of problems. Next, T-DepBM is proposed in the later section as well as the analysis of algorithms' computational complexity. Finally, simulation of T-DepBM and its results are show in the Section 4.4.

4.2 ILLUSTRATION OF PROBLEM

This chapter solves the problem of calculating coalitional value using the concept of budget. According to agents' cooperative behaviour, agents tend to seek others to collaborate in order to achieve common goals. They also tend to share information among each other to ensure transparency of communication and profit earned.

In the real world, activities such as conference, department activities, events and others are driven by the budget or funds. There are also other factors to be considered such as the mutual trust, value over functions and others. However, budget has been chosen as the main evaluation criteria for the proposed T-DepExp systems. This constraint is chosen because it allows an agent to maintain the minimum cost used in CF.

In this section, the problem is described using a case study of the furniture product line involving plywood, furniture and forestry companies. A furniture company $comp_{req}$ requests the plywood company $comp_{prod}$ to supply plywood in order to achieve its goal g_{req}. The goal g_{req} denotes company $comp_{req}$ is trying to maximize its profit by manufacturing furniture using a lower cost plywood supplied from company $comp_{prod}$. However, in order for both companies to collaborate, company $comp_{prod}$ demands company $comp_{req}$ to find forestry companies $comp_x$ to get a lower price for plywood supplies. The company $comp_x$ will provide tree logs for company

$comp_{prod}$ to manufacture the required plywood.

Let's say there are n numbers of forestry companies, competition between these companies is intensified when the cost of collaboration is increasing. Company $comp_{req}$ will request a price quotation from $comp_{x \in n}$ while searching for the lowest cost for woods. Company $comp_{x \in n}$ is able to request other forestry companies to supply extra tree logs if $comp_{x \in n}$ has insufficient supply of tree logs. Before doing so, the proposing furniture company $comp_{req}$ will allocate budget for purchasing tree logs. The company $comp_{prod}$'s goal g_{prod} is to earn a profit by supplying plywoods to company $comp_{req}$. When the company $comp_{req}$ requests tree logs from companies $comp_x$, the dependence relationships may consists of dep_{1to1}, dep_{or} and dep_{1to1}. The dep_{and} relationship describes the company $comp_{req}$ demands every forestry companies involved in the relationships to supply the required tree logs. The dep_{or} relationship denotes the company $comp_{req}$ needs to decide one of the forestry companies to supply the amount of tree logs it requested. The dep_{1to1} relationship denotes company $comp_{req}$ only has a company $comp_x$ to request for plywood supplies. The company $comp_{prod}$ is required to evaluate these relationships and tries to maximize its profit by minimizing the overall cost. Criterias to be considered during the evaluation process includes goal, budget and cost. Figure 4.1 illustrates the transitive dependence relationships between three companies.

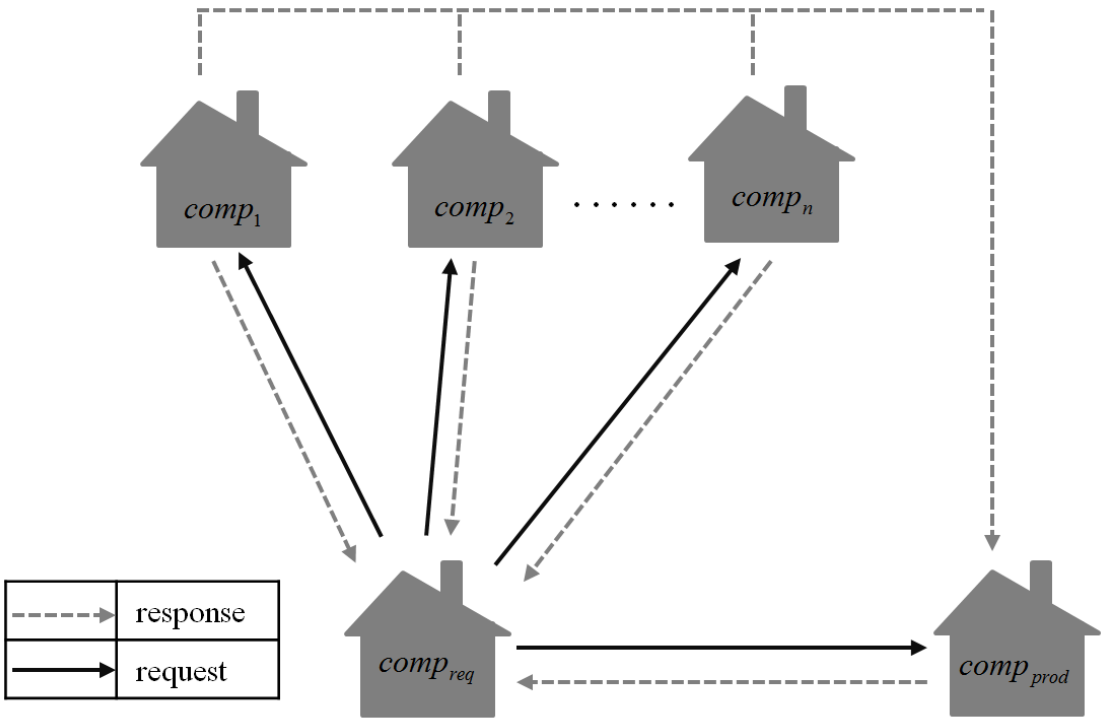


Figure 4.1 The T-Dep relationships between three companies

Example 3.1: Company $comp_2$ is the furniture manufacturer and it wants to produce furnitures to earn profit. Company $comp_2$ will negotiate with company $comp_1$ to settle agreement on the plywood supplies. Company $comp_1$ will request company $comp_2$ to find supplies for tree logs in order to get discounted price for plywoods. Hence, it is required to find plywood companies to form cooperation by surveying their production's price. Given $comp_2$ has communicated with set of companies $\{comp_3, comp_4, comp_5\}$, it will need to determine the lowest price for tree logs supplies. The company $comp_3$ possess a dep_{and} relationship with $comp_6$ and $comp_7$ while the $comp_4$ and $comp_5$ can supply tree logs independently (S-Autonomous). The cost for each dependence relationships can be denoted as $<comp_3, 50>$, $<comp_4, 40>$ and $<comp_5, 46>$.

Figure 4.2 illustrates the T-Dep relationships between agents based on the Example 3.1:

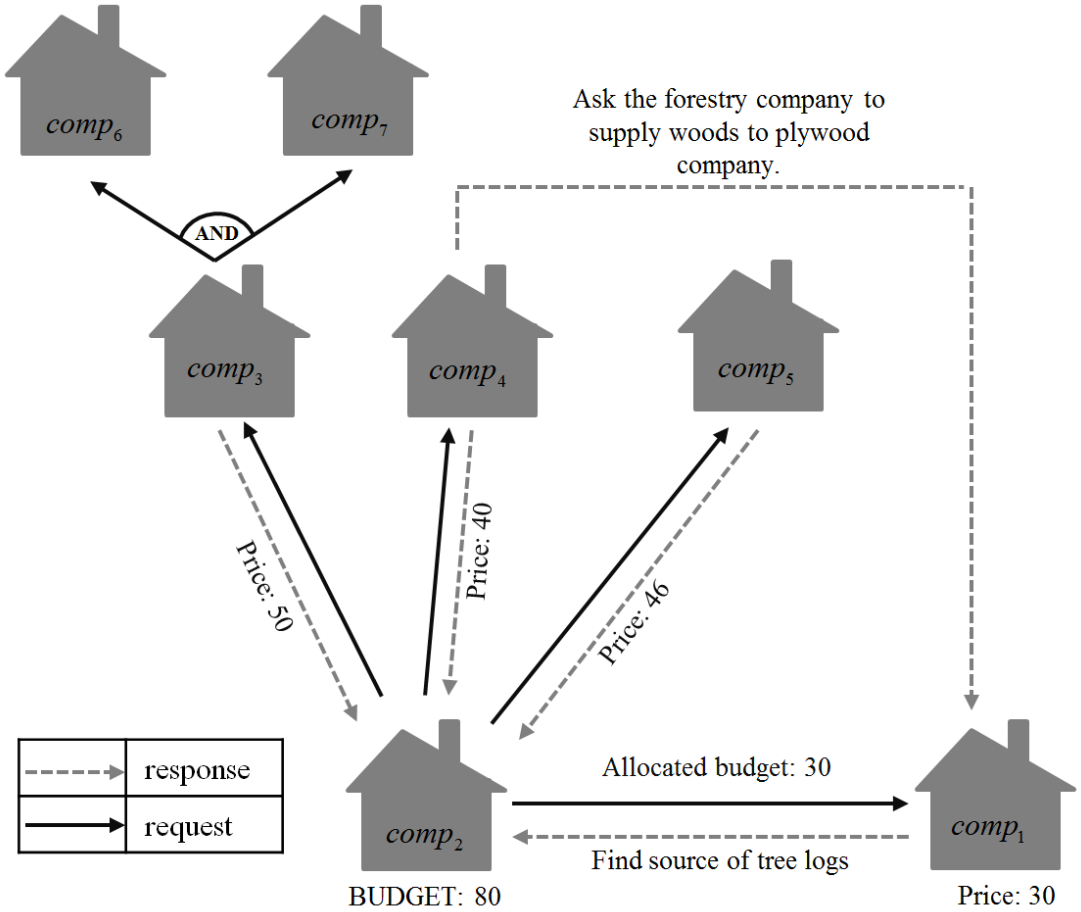


Figure 4.2 The agents' relationships based on the Example 3.1

The T-Dep relationships in Figure 4.2 can be expressed as:

$$dep = TDep(comp_2, comp_1, depchain)$$

where $depchain = comp_2 \xrightarrow{furniture} comp_4 \xrightarrow{woods} comp_1$. This shows company $comp_2$ will always search for an optimal cost based on its possessed budget. The budget in the Example 3.1 is set to be 80. Based on the Figure 4.2, profit for company $comp_2$ earned is 10 by deducting company $comp_1$'s budget from the cost of

company $comp_2$.

4.3 APPROACH

In this section, mathematical notations are presented first followed by T-DepBM's algorithm. Later on, the worst case scenarios are outlined based on the computational complexity analysis. It is to provide a lower bound guarantee for the proposed algorithms in T-DepBM systems.

Theorem 4.1: Total ratio of the dependence relationships in proposed coalition is 1.0.

Proof: According to three probability axioms based on Kolmogorov's Probability Calculus (Ash, 2012), total ratio of the dependence relationships can be denoted as 1.0. There are three types of dependence relationships to be considered and it is composed of the ratio of dep_{and} relationship P_{and}, dep_{or} relationship P_{or} and dep_{1to1} relationship P_{1to1}. The notation of total dependence relationships' ratio in a coalition is denoted as:

$$\sum_{i=1} P_{and} + \sum_{i=1} P_{or} + \sum_{i=1} P_{1to1} = 1.0 \tag{4.1}$$

The representative agent agt_{lead} needs to decide among these three relationships for choosing agents with an optimal cost to form a coalition.

Example 4.2: Consider a coalition consists of 10 agents led by a coalition representative agt_{lead}. The relationships in the coalition are composed of two dep_{or}, one dep_{and} and dep_{1to1} relationships.

Figure 4.3 illustrates the example of dependence relationship inside a coalition based on the Example 4.2:

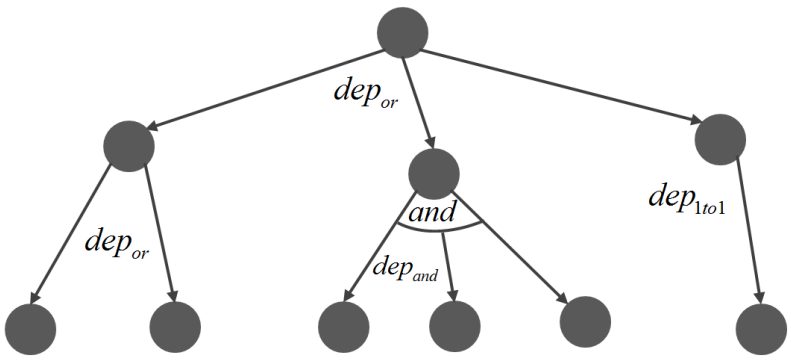


Figure 4.3 The coalition based on the Example 4.2

Equation 4.1 denotes the total ratio of the dependence relationships is $\sum_{i=1} P_{\{and,or,1to1\}} = 1.0$. The total ratio of the AND dependence relationship is denoted as $\sum_{i=1} P_{and} = 0.5$ as majority of the agents are involved in the dep_{and}. The $\sum_{i=1} P_{or}$ and $\sum_{i=1} P_{1to1}$ are 0.25 for the dep_{or} and dep_{and} relationship respectively.

4.3.1 Transitive Dependence based Budget Mechanism

In this section, T-DepBM is proposed to help an agent agt_{root} searches for an optimal coalition structure using a heuristic approach. The cooperative behaviour of a child agent agt_i allows information to be forwarded through parent agents until it reaches the agent agt_{root} during the CSG. In this chapter, the degradation of information is not under consideration and agents are assumed to be honest with each other due to their cooperative behaviour.

The algorithm for T-DepBM starts with a root agent agt_{root} seeking for lower cost dependence relationship through the i number of agents to form a coalition. Agent agt_{root} needs to calculate the estimated coalition cost required for forming a coaltion with the consideration of budget β_{root} by using the GET_COST() function. As stated in T-Dep relationships, an agent agt_i's estimated cost is passed transitively to root agent agt_{root} for calculation. The function will return a path with the least cost if the cost is lower than the budget and lowest cost. The path contains the list of the relations that has the least estimated coalition cost. Figure 4.4 shows the algorithm for agent agt_{root} checking the CF's validity with its budget.

ALGORITHM 4.1. The agent agt_{root} checking the validity of the coalition cost.

```
Function  : GET_FEASIBLE()
Input     : Root_Agent, budget_RootAgent, Expected_pay, counter
Output    : path
IF counter < 5 THEN
      CoalitionStatusMessage = GET_COST(Root_Agent, path, budget_RootAgent)
     IF CoalitionStatusMessage IS valid THEN
          RETURN CoalitionStatusMessage.get_path()
     ELSE
          counter = counter + 1
          DISPLAY CoalitionStatusMessage.fail_message()
          CHANGE _CRITERIA()
          GET_FEASIBLE()
ELSE
    RETURN CoalitionStatusMessage.exceed_attempt()
```

Figure 4.4 Algorithm for agent agt_{root} retrieving the status of CF

Subsequently, the function GET_COST() will return a message about the validity of

path and coalition cost. If coalition cost is lower than the projected budget as described in Condition 2.4, path will be passed to root agent agt_{root} to form a coalition. If the coalition cost exceeds budget, a failure message will be generated and passed back to agent agt_{root}. It will revise its plan to choose a different path. The number of attempts to change criteria will be limited up to five times. The agent agt_{root} will seek for alternate solutions to achieve its goals if it failed every attempt.

Further details on the function GET_COST() algorithm is shown in Figure 4.5. It describes the root agent agt_{root}'s procedure for evaluating different types of dependence relationships. There are three dependence relationships considered for T-DepBM which consists of (1) SINGLETON relationship dep_{1to1}, (2) AND relationships dep_{and} and (3) OR-relationships dep_{or}. Figure 4.5 shows the Algorithm 4.2 which consist of agent agt_{root}'s evaluation on the dependence relationship.

ALGORITHM 4.2. Analyzing the dependence relationship.

```
Function : GET_COST()
Input    : current_agent, path, coalition_budget
Output   : message
  IF path DO NOT CONTAIN current_agent THEN
      IF current_agent DO NOT HAVE any children THEN
          IF current_agent_cost WITHIN coalition_budget THEN
              ADD current_agent INTO path
              RETURN success_message
          ELSE
              RETURN not_feasible_message
      ELSE
```

(See next page)

```
            FOREACH relationship IN current_agent DO
        relationshipType = ANALYSE_RELATIONSHIP ( current_agent )
                IF relationshipType IS "single-relationship" THEN
                        SINGLE_RELATIONSHIP()
                IF relationshipType IS "and-relationship" THEN
                        AND_RELATIONSHIP()
                IF relationshipType IS "or-relationship" THEN
                        OR_RELATIONSHIP()
        ENDFOR
ELSE
    RETURN exist_path_message
```

Figure 4.5 The GET_COST() algorithm

The overall concept in Algorithm 4.2 is the root agent agt_{root} will try to evaluate the agent agt_i's dependence relationships based on the validity of the coalition cost U_i. The concept of path is derived from graph theory where agents traverse through their relationships. If there exists a child agent agt_{chd}, it will continue to traverse until it reaches the leaf agents. The dependence relationship will be further break down using the GET_COST() function recursively. The agent agt_{root} will receive final coalition cost and budget information from child agents while the algorithm traverse through the recursive dependence relationships.

4.3.2.1 Singleton Dependence Relationships

The SINGLETON relationship shows an agent agt_{root} only has a single agent to depend on for the required capabilities. There is no alternate choice for the root agent agt_{root} and it has to depend on agent agt_i to achieve its goal. Figure 4.6 shows the algorithm for evaluation of dep_{1to1} coalition cost.

ALGORITHM 4.3. Evaluation of the dep_{1to1} relationship

Function : SINGLE_RELATIONSHIP()
Input : *current_agent*, *path*, *coalition_budget*, *expected_pay*
Output : *message*
IF *current_agent* ADD INTO *path* IS *feasible* **THEN**
nextBudget = *budget* – *current_agent.actionCost*
ADD *current_agent* INTO *path*
temp_message = GET_COST(*child_agent*, *path*, *nextBudget*, *expectedPay*)
IF *temp_message* IS *success* **THEN**
RETURN *success_message*
ELSE
RETURN *not_feasible_message*
ELSE
RETURN *not_feasible_message*

Figure 4.6 Algorithm for evaluating the dep_{1to1} relationship

The evaluation of a single agent's coalition cost is rather straight forward as it only includes agent agt_i into budget and evaluates accordingly. If the agent agt_i's cost of performing its capabilities is within the budget range, the dep_{1to1} relationship is consider as valid. Subsequently, the latest budget will be updated by including agent agt_i's cost. Otherwise, a "*not_feasible_message*" message will be returned to the parent agent agt_{prt} where it needs to choose an alternate route.

4.3.2.2 OR-Dependence Relationships

The OR-Dependence relationship shares some common characteristics with the SINGLETON relationships such as it requires one child agent agt_{chd}'s capabilities. It is possible to choose more than one agent for the required action but it depends on

the child agents' capacity of supplying their capabilities. In our proposed model, we assume one agent is sufficient to achieve the goal by its capabilities. However, the difference between these two relationships is dep_{or} relationship requires to evaluate every agent agt_i's cost involved before concluding the feasibility of the coalition cost. Figure 4.7 shows the algorithm for dep_{or}'s evaluation:

ALGORITHM 4.4. Evaluation of the dep_{or} relationship

```
Function  : OR_RELATIONSHIP()
Input     : current_agent, path, coalition_budget, expected_pay
Output    : message
 IF current_agent ADD INTO path IS feasible THEN
        Candidate = CHOOSEAGENT(current_agent, expected_pay)
        nextBudget = budget – current_agent.actionCost
        ADD current_agent INTO path
        temp_message = GET_COST(Candidate, path, nextBudget, expectedPay)
        IF temp_message IS success THEN
                RETURN success_message
        ELSE
                RETURN not_feasible_message
 ELSE
      RETURN not_feasible_message
```

Figure 4.7 Algorithm for evaluating the dep_{or} relationship

The evaluation process starts by choosing a candidate which possesses a temporary least cost. Subsequently, every agent engaged in the dep_{or} relationship will be evaluated in a sequential manner. The child agents of each candidate will be considered for the evaluation as well since accumulated cost is essential for determining a "least cost" path. It is presented in the function CHOOSEAGENT()

and the cost will be recursively passed back indirectly to the parent agents until it reaches the agent agt_{root}.

4.3.2.3 AND-Dependence Relationships

The evaluation of AND-dependence relationships requires every agent agt_i's capabilities in order for the agent agt_{prt} to achieve its plan. The total cost of the AND-Dependence relationship must be lower than the agent agt_{root}'s budget. If the sum of agent's cost exceed the threshold of the β_{root}, dep_{and} relationship will not hold and the agent agt_{root} have to revise its budget and seek for an alternate path. Figure 4.8 shows the algorithm for evaluating the dep_{and} relationship.

ALGORITHM 4.5. Evaluation of the dep_{and} relationship

```
Function  : AND_RELATIONSHIP()
Input     : current_agent, path, coalition_budget, expected_pay
Output    : message
 IF current_agent ADD INTO path IS feasible THEN
     ADD current_agent INTO path
     nextBudget = budget – current_agent.actionCost
     WHILE have_child(current_agent) DO
         temp_msg = GET_COST(get_child(current_agent), path, nextBudget,
         expectedPay)
         IF temp_msg IS valid THEN
             num_appropriate = num_appropriate + 1
     END_WHILE
     IF num_appropriate EQUALS getNumChild(current_agent)THEN
         RETURN success_message
     ELSE
```

(See next page)

```
            RETURN not_feasible_message
ELSE
    RETURN not_feasible_message
```

Figure 4.8 Algorithm for evaluating the dep_{and} relationship

The Algorithm 4.5 starts with evaluation of child agent agt_{chd}'s cost. If it is not a leaf agent agt_{leaf}, the algorithm will continue to traverse until it reaches one. The accumulated agents' cost $\sum_{i=1}^{n} U_i$ will be calculated and passed to the agent agt_{root} for evaluating the status of relationships.

4.3.2 Computational Complexity Analysis

The efficiency of T-DepBM is calculated based on the computational time complexity. Consider worst case of possible relationships occurred in CF is ${}^{n}P_k = \frac{n!}{(n-k)!}$ where the k is permutations set of n. The first step in Algorithm 1 is agent agt_{root} proposes a feasible coalition by calculating cost of the coalition. The complexity of GET_COST() function is O($({}^{n}P_k \times n)\log_i(n)$) by considering worst possible outcome. It only occurs when every relationship in the society is dep_{and}, $P_{and} = 1.0$ which results in the computational complexity of O($\log_i n$). If the coalition only consists of the dep_{or} and dep_{and}, the complexity would be O($({}^{n}P_k \times n)$). Accounting for every possible worst case of T-DepBM, the computational complexity of these algorithms are O($\phi \times (({}^{n}P_k \times n)\log_i(n))$) where ϕ is the number of revision on budget or evaluation element until it reaches terminating conditions. It can be feasible cost have been derived or a failure of the CF (Note: For further calculation of the algorithm complexity, see Appendix C).

Worst case has been detailed which the big O notation has covered the computational complexity of CF. The algorithm guarantees a computational complexity upper bound of $\mathrm{O}((^{n}P_{k} \times n)\log_{i}(n))$. The computational complexity can be further reduce if an upper bound has been applied on the maximum agent in the coalition. However, the proposed T-DepBM's algorithm does not ensure the optimal result in the CF`.

4.4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the T-DepBM has been implemented into the T-DepExp System to simulate its effectiveness of forming T-Dep based coalition through the cooperative agents. In the first subsection, the parameters of the experiments are introduced followed by the performance measurements. Subsequently, the three experiments are conducted and results are discussed.

4.4.1 Parameters and Performance Measurements

Based on the authors' knowledge, there is no standard database available for benchmarking. Hence, the dataset used in this experiment is randomly generated in real time and the experiment is conducted 10,000 times to obtain accurate result. The performance measurements of the T-DepExp systems are the total coalition cost, $\sum_{i=1}^{n} U_i$ and the CF's profit $\sum_{i=1}^{n} R_i$. These two measurements decide the worthiness of CF organized by an agent agt_{root}. Several simulation parameters are used to generate the environment of MAS society to conduct T-DepBM's experiment. Parameters consists of number of agents, probability of the dependence relationships generated

as well as agent agt_{root}'s budget. The following Table 4-1 lists the simulation parameters:

Table 4-1 The simulation parameters of the T-DepExp systems

Parameters	***Descriptions***
n	Agents' number in the society that has the intention to join the coalition.
P_a	The probability of an agent having dependence relationships with others.
P_{and}	The dep_{and}'s ratio received by the agent agt_{root} during CF.
P_{or}	The dep_{or}'s ratio received by the agent agt_{root} during CF.
β_{root}	The budget for the agent agt_{root} to form the coalition.
σ_i	The grading scale of the plan priority for the agent agt_i to achieve.
cp_i	The grading scale of the capabilities for the agent agt_x's usage for joining the coalition.

These experiments aim to study the effect of agent's number, relationship ratio, dep_{and}'s ratio and budget quota corresponding to coalition cost and profit. The T-DepExp systems is developed based on JAVA to simulate CF of cooperative agents. Experiments are conducted on a workstation with the following specifications: *Intel Xeon Processor*, 16*GB RAM* and *Nvidia Quadro*. If the result of T-Dep based CF is $\sum_{i=1}^{n} R_i = 0$, it implies there are cycles or "deadlock"s occurs inside the CF. A similar result will be obtained if coalition cost exceeds the projected budget.

4.4.2 Experiment 4.1: Dependence Ratio

The first experiment aims to study the dependence ratio P_a corresponding to total cost required to form the coalition. Experiment 4.1 uses the following parameters:

$$[\beta_{root} = 450, P_a = \{0.2, 0.4, ..., 2.0\}, n = 450, P_{and} = \{0.2, 0.4, ..., 1.0\}]$$

The simulation result is presented at Figure 4.9 and increasing P_a has rapidly decreased the profit CF earned by agents involved in CF. Every relationships, dep_{and} with various ratio has reduced coalition cost significantly as the coalition formed for feasible profit does not hold. When the P_{and}=1.0, the coalition total costs shows "0" and represents the failure of CF at P_a=1.0. There is a significant slope for the total costs that occurs from P_a=1.0 and P_a=1.2. The increment of P_a has increased number of dependence relationships during CF. This causes the coalition costs to decrease where feasibility of coalition is not guaranteed during CF.

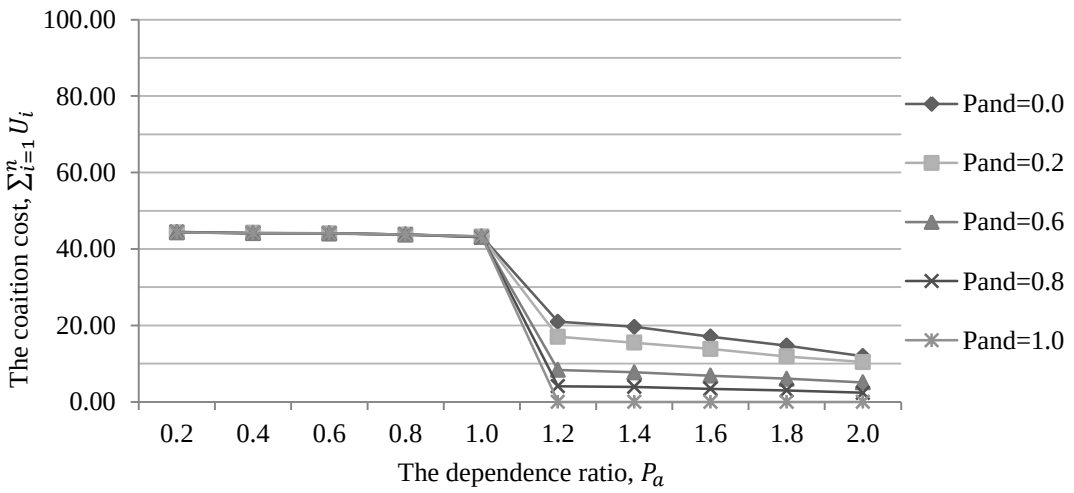


Figure 4.9 The dependence ratio corresponding to the coalition total cost

The observation from the Figure 4.9 has shown that CF has a higher tendency to fail if the dep_{and}'s and P_a's ratio is high. Agent agt_{root} is required to consider more agents into coalition and leads to total cost easily exceed β_{root}. Hence, feasible coalition is easier to formed if the ratio of dep_{and} relationship is less than P_a=1.0.

4.4.3 Experiment 4.2: Agents' Number

The second experiment aims to study the impact of agent's number on total cost of coalition. Experiment utilizes the following parameters:

$$[\beta_{root} = 450, P_a = 1.0, n = \{50,100,...,500\}, P_{and} = 0.6]$$

The simulation's result of Experiment 4.2 is presented in Figure 4.10.

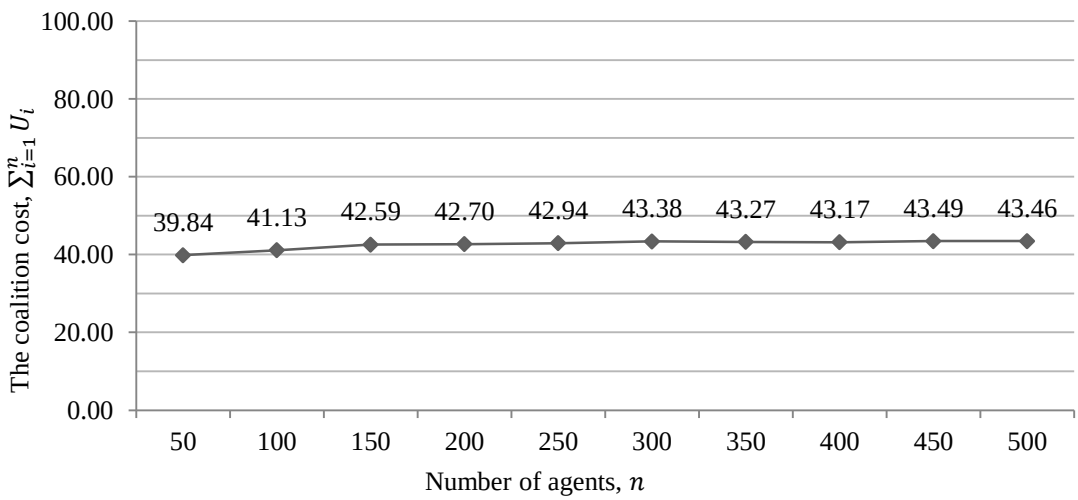


Figure 4.10 The agent's number corresponding to the coalition cost

Based on observation, increasing the number of agents does not increase coalition cost significantly. The total coalition cost has increased in a very small amount which

is from $\sum_{i=1}^{n} U_i$ =39.84 to 43.36. One of the reasons for small increase is agent agt_{root} has more choices with the emergence of more agents with the required capabilities.

However, increasing the number of agents shows less profit is earned because traversing through more agents cost more. The worst case of the dependence relationships formed with n agents can be an exponential increase in coalition cost where a lower budget will not be sufficient for CF. Hence, the agents' number will decrease coalition profit because traversing the relationships require a higher budget to fulfil the common goals among agents.

4.4.4 Experiment 4.3: Budget

The third experiment aims to study the agents' projected budget corresponding to the coalition profit. The experiment uses the following parameters:

$$[\beta_{root} = \{50,100,...,600\}, P_a = 0.5, n = 400, P_{and} = 0.5]$$

Figure 4.11 shows the impact of agent agt_{root}'s budget on coalition total cost. Increasing the agent agt_{root}'s budget causes total coalition cost's growth rate in a concave down trend. The total coalition cost depends on number and type of relationships during the CF. When the budget allocated has reached its limit, the coalition cost will increase slowly. This is because the coalition formed has reached the state of agents charging same amount of cost for cooperation. The profit will be calculated after deducting the cost from the allocated budget.

Based on Equation 2.8, the coalition profit can be obtained by deducting actual

coalition costs from the budget. Figure 4.12 shows agent agt_{root}'s budget corresponding to the coalition's profit.

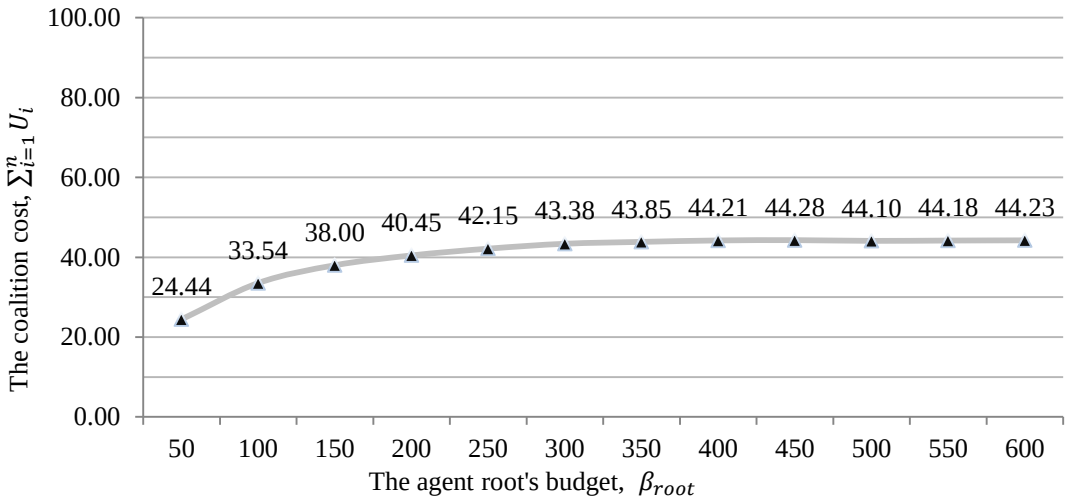


Figure 4.11 The agent agt_{root}'s budget corresponding to the coalition cost

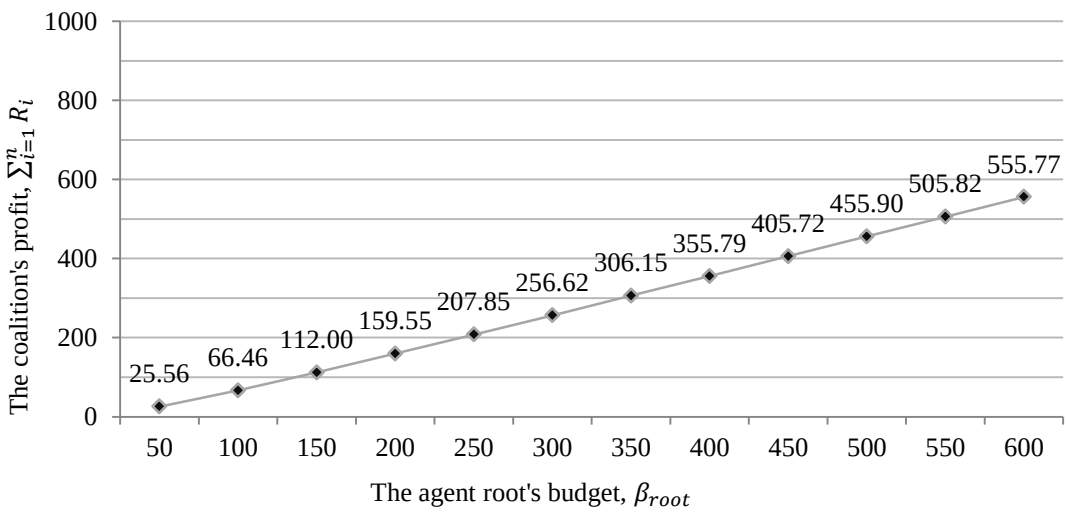


Figure 4.12 The agent agt_{root}'s budget corresponding to the coalition's profit

Based on observation, the growth rate of the profit is increasing in a linear rate. The incremental rate denotes the coalition cost is not significant given the budget is sufficient. However, it might cause the incident of abusing budget in order to earn

high amount of profit through collaboration. We assumed the agents are honest and always provide accurate information to others.

4.5 SUMMARY

In this chapter, T-DepExp system has been proposed to help calculate the coalition cost and profit. It utilizes the concept of budget to perform searching for an ideal CSG. T-DepExp system has showed the number of agents from $n = 50$ until 500 does not increase coalitional cost as expected. Besides, the increasing agent agt_{join} has increased coalitional profit linearly and coalition members are able to gain more profit during collaboration.

Chapter 5 Verification of Agents' Relationship Validity

5.1 INTRODUCTION

This chapter proposes a knapsack based dependence relationship validation mechanism (KDRVM) to aid agents validate their dependence relationship during CSG. KDRVM determines the validity of the dependence relationship between agents in a coalition. The valid dependence relationships ensures coalition profit through the checking process of each relationship's cost by using the concept of solving a 0-1 knapsack problem. The 0-1 Knapsack problem (Dantzig, Mazur, & Mazur, 2007) is a combinatorial optimization problem that aims to maximize value while trying to minimise constraint such as weight. According to Kellerer, Pferschy, and Pisinger (2004), general notation of 0-1 knapsack problems is denoted as $\sum_{h=1}^{n} pr_h x_h$ while satisfying the constraint $\sum_{h=1}^{n} w_h x_h \leq W$. The w_i represents weight of an object h while pr_h represents the profit of the object h. The maximum weight is denoted as W. The knapsack problem has offered a model for constraint decision making which has been implemented into CF (An et al., 2007) due to its practical importance.

There are several well-known derivatives of knapsack problem presented in Lagoudakis (1996)'s survey such as bounded knapsack, unbounded knapsack, multiple choice 0-1 knapsack, multiple choice knapsack and others. The implementation of bounded knapsack problem is common in the distributed problem

solving approach. For example, An et al. (2007) has demonstrated a CSG with the bounded knapsack based on their proposed transitive dependence relationship. The transitive dependence relationship based algorithm help to decide which agents during CSG. On the other hand, the unbounded knapsack problem has been applied to resolve the coalition overlapping issues in some papers (Yusen, Jun, Chongjun, & Junyuan, 2012; Zick, Chalkiadakis, & Elkind, 2012). This is similar to the unbound knapsack problem where the upper bound of coalition number an agent can join is not limited. Moreover, the knapsack problem's paradigm has gained its reputation in the decision making of robotic CF's decision making. Several techniques to answer the decision making in the robotic based on the knapsack problem has been developed such as quantum genetic (Han & Kim, 2000; Y. Xiong, Chen, Miao, & Wang, 2004), particle swarm optimization (Juang, 2004; Wang, Diao, & Gao, 2008) and ant colony optimization algorithms (W. q. Xiong, Zhou, & Wei, 2005).

In this chapter, the bounded 0-1 knapsack problem has been implemented into KDRVM to decide the dependence relationship's validity. The communication rate between agents in coalition is assumed to be proportional to Metcalfe's law (Shapiro & Varian, 1999) during verification process. This chapter is organized as the following sequences: First, mathematical notations inside KDRVM are introduced followed by the problem statement. Subsequently, the algorithm for designing KDRVM is presented and described. The last section consists the summary of our methodology.

5.2 MATHEMATICAL NOTATIONS

The 0-1 Knapsack problem is used as a anology for us to study the calculation of

dependence relationship's feasibility. The following notations are definitions for the main elements inside KDRVM.

Definition 5.1: Agent's value v represents the cost of an agent to perform an action or share its resources.

The agent agt_{chd}'s value denotes the cost to help parent agent agt_{prt} by supplying its capabilities. An agent's value v is generated during CVC and bounded with numerical value of $1 \le v \le 100$.

Definition 5.2: Maximum capacity φ represents the maximum weight a root agent agt_{root} can take.

The maximum capacity serves as a threshold for agent agt_{root}'s limit on weight. The maximum capacity is determined by agent agt_{root}'s expectation corresponding to the number of agents in a coalition. If total weight is higher than the maximum capacity, the dependence relationship will be marked as not valid.

Definition 5.3: Current capacity c represents the current weight during traversing dependence relationships.

The current capacity is the accumulative weight of agents when agent agt_{root} is traversing down the dependence relationships in a coalition. It will be updated as more agents have been explored by agent agt_{root}. It is denoted as:

$$c = \sum_{r=1}^{j} \sigma_r \tag{5.1}$$

where j represents the number of agents involved in relationships while the σ_r represents the weight of agent $agt_r \in Agt_j$ in the $\{\sigma_1, \sigma_2, ..., \sigma_j\}$.

5.3 PROBLEM STATEMENTS

The dependence relationships can be modelled into a dependence graph for visualizing agents' relationships (J. S. Sichman & Conte, 2002). One of the problems encountered during CSG is the validity of agents' relationship. If dependence relationship is not valid, the stability of a coalition will not hold. There are constraints to be considered during the evaluation of agent's relationship such as agents' weight and value. To ensure a coalition stability, the total weight of dependence relationship $\sum_{r=1}^{j} \sigma_r$ should not exceed agent agt_{root}'s maximum capacity φ. Otherwise, coalition stability will be affected and agents will not gain any profit through CF.

Consider the scenario where an agent agt_{root} wants to evaluate its relationships in a coalition z. The coalition z consists of n agents which includes agent agt_{root}. It is considered as leader of the coalition as it want to ensure the coalition's profit. The other members are denoted as coalition members or set of child agents $Agt_x \in z$. The basic dependence relationship between two agents is represented by $basic\_dep(agt_{root}, agt_j, g_k, p_{gk})$. This shows agent agt_{root} is having relationship with agent agt_j under the plan p_{gk} and aims to achieve the goal g_k. The dependence graphs and types of the relationships to be validated are illustrated in Figure 5.1:

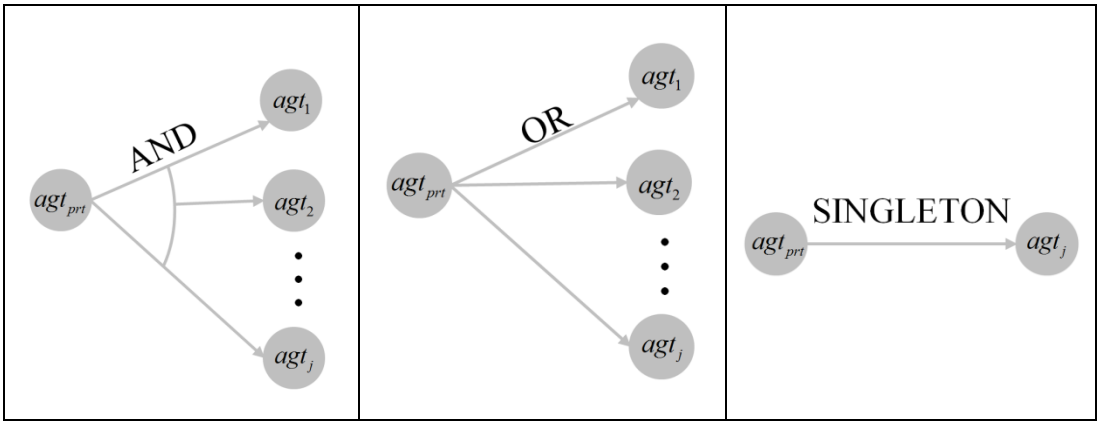


Figure 5.1 Type of dependence relationships to be validated

Figure 5.1 denotes the relationships of dep_{and}, dep_{or} and dep_{1to1} respectively from left to right. Agent agt_{root} will evaluate these relationships to determine the validity of relationships. The dep_{and} and dep_{or} consists of j number of child agents while the dep_{1to1} only has one agent to depend on.

Consider the following example: In a coalition z, agent agt_{root} has a common goal g_{root} to achieve. It depends on three agents using dep_{or} that are agt_1, agt_2 and agt_3 respectively. For agent agt_1 to achieve g_{root}, agt_1 requires all three agents' capabilities that are agt_4, agt_5 and agt_6. The agt_1's total weight of the dependence relationship is $\sum(\sigma_{agt4}, \sigma_{agt5}, \sigma_{agt6})$=42 and it has exceed agent agt_{root}'s maximum capacity. Hence, the relationships under agent agt_1 is shown to be not valid. If agent agt_{root} choose agent agt_2 to depend on then this is a singleton relationship dep_{1to1}; provided agt_1 and agt_3 are not chosen. Agent agt_2's total weight is σ_{agt2}=35 and it does not have any child agent. At current state, agent agt_2 is one of the candidates for relationships in the stable coalition. However, it is not guaranteed an optimal result for the total weight. The third agent agt_3 has dep_{or} with agents agt_7 and agt_8

under the agt_3's goal g_3. Agent agt_3's total weight is $\sum(\sigma_{agt3}, \sigma_{agt7})$=25 and $\sum(\sigma_{agt3}, \sigma_{agt8})$=31 respectively. Agent agt_3 is another possible candidate for a valid relationship. The example above is illustrated in Figure 5.2:

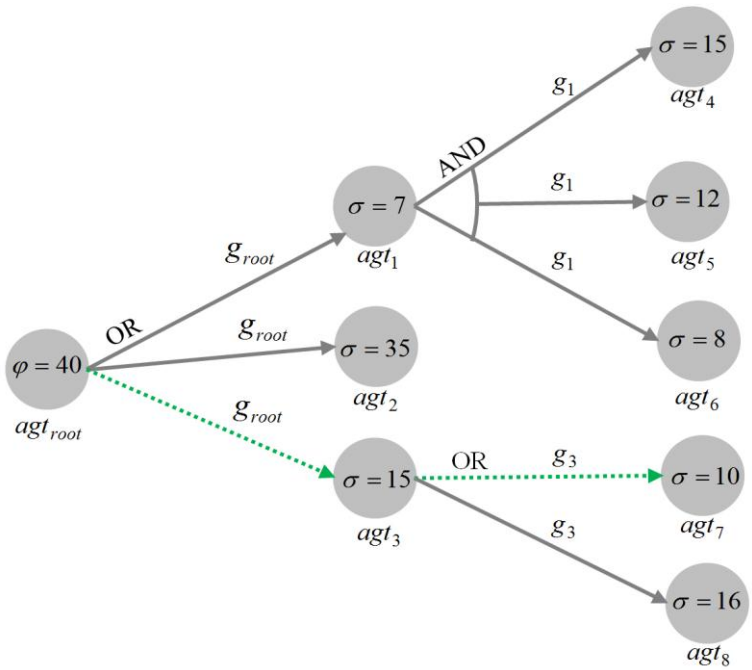


Figure 5.2 Example of the dependence relationship in a coalition

Based on Figure 5.2, an agent agt_{root} needs to make a decision to choose the ideal relationships to achieve its goals while maximizing profit. There are two candidates for agent agt_{root} to depend on which are agent agt_2 and agt_3. If agent agt_{root} wants a simple relation with fewer agents, it will depend on agent agt_2. Given the weight of coalition has a higher priority, agent agt_{root} will depend on agent agt_3 and agent agt_7 to achieve its goal.

5.4 APPROACH

The following are the methodologies used to verify the validity of the relationships.

First, the concept of knapsack problem is explained followed by the introduction of KDRVM's evaluation algorithms. Lastly, the computational complexity of algorithms are presented and explained.

5.4.1 Knapsack based Dependence Relationship Validation Mechanism

The KDRVM uses depth-first search and greedy algorithms in traversing the dependence graph while requesting for agents' information. The information includes weight and value of the particular child agents for agent agt_{root} to evaluate. The KDRVM aims for generating an optimal result where the profits of coalition are guaranteed. Figure 5.3 shows the algorithm for KDRVM:

ALGORITHM 5.1. Calculate the validity of the root agent's dependence relationship

Function : evaluateDepRelationship()

Input : $agt_{root}, \varphi_{root}$

Output : *validity_of_relationship*

validity = *true*

$agt_{crt} = agt_{root}$

WHILE have_child(agt_{crt})**DO**

  IF have_child(agt_{crt})**THEN**

    IF get_history(agt_{crt})IS *valid* **THEN**

      $dep =$ identify_relationship (agt_{crt})

      IF dep IS dep_{or} **THEN**

        or_dependence()

      ELSE IF dep IS dep_{and} depand **THEN**

        and_dependence()

      ELSE IF dep IS dep_{1to1} dep1to1 **THEN**

        (See next page)

        sing_dependence()

```
                ELSE
                        terminate( agt_crt )
                        agt_crt = stack_remove()
                        update(parent_agent, weight)
        ELSE
            IF check_validity() IS invalid THEN
                        terminate()
            ELSE
                        update(parent_agent, weight)
        update_information()
ENDWHILE
```

Figure 5.3 Calculate the validity of the root agent's dependence relationship

The KDRVM starts by assigning agent agt_{root} as the current agent agt_{crt}. Next, KDRVM will evaluate the status of agent agt_{crt}'s child agents. If agent agt_{crt} has one or more child agents, the agent agt_j's dependence relationship will be further analysed. Otherwise, a terminate signal will be passed to parent agent of the agent agt_{crt} to inform end of relationships. Upon successful evaluation of the child agents, existence of current agent will be searched and checked in the history function. It is based on the concept of stack and have similar properties such as first in and first out (FIFO). List of agents will be stored in the history while agent agt_{root} traverses through the dependence graph. It is to prevent repeating of agents in dependence relationships that might result in unresolved claim of actions or resources. It offers backtracking features for relationships while it searches for an optimal solution. In addition, agent agt_{root} is able to keep track of the path with the backtracking features. The following Algorithm 5.2 shows the history function of KDRVM.

ALGORITHM 5.2. Evaluate presence of agent agt_{crt}'s history

```
Function  : get_history()
Input     : agt_crt
Output    : validity
IF agt_crt IS valid THEN
    IF agt_crt IS NOT visited THEN
        stack_add( agt_crt )
        validity = true
    ELSE
        stack_remove()
ELSE
    stack_remove()
RETURN validity
```

Figure 5.4 The evaluation of agent agt_{crt}'s history

If agent agt_{crt} is not present in the history, KDRVM will continue to evaluate the types of the dependence relationship. There are generally three types of the dependence relationships to be validated which are dep_{or}, dep_{and} and dep_{1to1} as shown in Chapter 2. The algorithms have been divided into three parts in KDRVM.

5.4.2 Evaluation Algorithm according to the Dependence Relationships

There are three types of dependence relationships that need to be verified in the coalition. Each dependence relationship's characteristic have been analysed and implemented into KDRVM as follows:

5.4.2.1 Singleton-Dependence

The first dependence relationship to be validated in KDRVM is dep_{1to1} relationship. The evaluation of the dep_{1to1} relationship is shown in Algorithm 5.3:

ALGORITHM 5.3. Evaluate the dep_{1to1} relationship
Function : sing_dependence()
Input : agt_{crt}
Output : *validity*
agt_{crt} = *agent*
update(*current_capacity*, *value*(agt_{crt} , c))
IF $c \leq \varphi$ **THEN**
dep_{1to1} = valid
ELSE
dep_{1to1} = invalid
RETURN *validity*

Figure 5.5 The evaluation of the dep_{1to1} relationship

It starts by denoting the only child agent agt_j as agent agt_{crt}. Subsequently, the record of child agent will be inserted into history and checks its record. After the verification process of agent agt_{crt}'s history, its weight will be added into the current capacity. After that, the child agents of agent agt_{crt} will be further evaluated until it reaches leaf agents. If current capacity does not exceed the agt_{root}'s maximum capacity, agent agt_{prt} will continue traverse until it reaches leaf agent.

5.4.2.2 OR-Dependence

The evaluation of the OR-Dependence involves every agent in dependence relationships to search for agent agt_{chd} with an optimal weight. However, the agent agt_{prt} can choose one particular child agent agt_{chd} to depend on as other child agents contain duplicated actions and resources. We assume one agent agt_{chd} is sufficient to

supply the required action or resources to the parent agent agt_{prt}. Figure 5.6 shows the dep_{or} relationship evaluation algorithm:

ALGORITHM 5.4. Evaluate the dep_{or} relationship
Function : or_dependence()
Input : agt_{crt}
Output : *validity*
FOR *agent* IN GET_LIST(dep_{or} , agt_{crt}) **DO**
agt_{crt} = *agent*
update(*current_capacity*, *value*(agt_{crt} , c))
ENDFOR
IF $c \leq \varphi$ **THEN**
dep_{or} = valid
ELSE
dep_{or} = invalid
RETURN *validity*

Figure 5.6 The evaluation of the dep_{or} relationship

The process of evaluating dep_{or} relationship starts with checking the weight of child agents in relationships. The current agent agt_{crt} 's weight will be added to the current capacity if it is lower than the maximum capacity. Also, this process involves every child agent agt_j in the relationships. During evaluation, child agent will replace agent agt_{crt} and continue until it reaches leaf agent. In the end, the total weight will be sent to parent agent to evaluate the validity of dep_{or} relationship.

5.4.2.3 AND-Dependence

The evaluation of AND-Dependence relationship considers every involved agents' weight during the evaluation process. If one of the agent agt_j's weight involved in dep_{and} relationship exceeds agent agt_{root}'s maximum capacity, it will terminate instantly. Figure 5.7 shows the evaluation of dep_{and} relationship:

ALGORITHM 5.5. Evaluate the dep_{and} relationship

Function : and_dependence()

Input : agt_{crt}

Output : *validity*

dep_{and} = valid

FOR *agent* IN GET_LIST(and, agt_{chd}) **AND** dep_{and} IS valid **DO**

update_best_offer(*agent*)

agt_{crt} = *agent*

update(*current_capacity*, *value*(agt_{crt}, c))

IF $c \le \varphi$ **THEN**

dep_{and} = valid

ELSE

dep_{and} = invalid

RETURN *validity*

ENDFOR

Figure 5.7 The evaluation of the dep_{and} relationship

The Algorithm 5.5 starts with agent agt_{crt} calculating the total weight of child agents utilizing function *update_best_offer*(). This function also checks upper bound of agent agt_{root}'s capacity while traversing the dependence relationship. Then the coalition value will be accumulated until it reaches leaf agents. It will terminate once it reaches leaf agents. The evaluation of dep_{and} relationship is based on comparing the current capacity of every agent and agent agt_{root}'s maximum capacity. If the

current capacity exceeds agent agt_{root}'s maximum capacity, the traversing process of relationship is terminated and marked as invalid.

5.4.3 Computational Complexity Analysis

In worst condition, the KDRVM will traverse or visit all agents. As discussed, the evaluation of relationship is based on the agents' weight and values are the second priority during the evaluation process. The KDRVM terminates when following conditions are satisfied:

- The agents Agt_j's total weight is higher than agent agt_{root}'s maximum capacity.
- The agent agt_j is visited twice in dependence relationship (a cycle exists).
- The KDRVM has finished with Agt_j's total weight less than agent agt_{root}'s maximum capacity.

The computational complexity of KDRVM is O(E^v) and it is NP-Complete (For more detail on the complexity analysis, please refer to Appendix D). The evaluation of dep_{and} relationship has a big impact on computational complexity as shown in Algorithm 5.1 and Algorithm 5.5 where both are O(E^v). The worst case for KDRVM is when every dependence relationships in the coalition consist of dep_{and} relationship. It is required to traverse all agents in the dep_{and} relationship. The evaluation of dep_{or} and dep_{1to1} relationships has a lower computational complexity compared to dep_{and} relationship because agent agt_{root} can choose a convenient agent agt_j with the smallest weight to cooperate with.

5.5 EXPERIMENTAL RESULTS

The performance measurements for experiments are the coalition value $\sum_{i=1}^{n} v_i$ and communication rate μ in the coalition. We have recorded down the valid dependence relationships where invalid ones are discarded. The $\sum_{i=1}^{n} v_i$ denotes the total profit of dependence relationship while considering agents' total value and weight. The communication rate μ shows the number of interaction between agents occurred during KDRVM's process. The interaction between agent agt_{root} and $agt_j \in Agt_j$ is recorded in a full communication response where the action of sending and receiving message from agents is counted as one interaction.

5.5.1 Hypothesis and Experimental Setup

5.5.1.1 Experimental Setup

The simulation of proposed KDRVM is conducted on the workstation with the following specification: *Dell Optiplex* 390, *Core i*3 *processor*, 4*GB RAM*, 7200*RPM HDD*. Moreover, the simulation is conducted with the iteration of 100 with a randomly generated dataset. The following Table 5-1 shows the input parameters for the simulation:

Table 5-1 Simulation parameters for the KDRVM

Parameters	***Description***
n	The number of agents in the coalition excluding agent agt_{root}.
φ_i	The capacity of the agent agt_{root}.

(Table 5-1 Continued)

$\sigma_{\min}$	The minimum weight of the agent $agt_j \in Agt_j$.
$\sigma_{\max}$	The maximum weight of the agent $agt_j \in Agt_j$.
$v_{\min}$	The minimum value of the agent $agt_j \in Agt_j$.
$v_{\max}$	The maximum value of the agent $agt_j \in Agt_j$.
P_{and}	The ratio of dep_{and} in the coalition.
P_{or}	The ratio of dep_{or} in the coalition.
P_{1to1}	The ration of dep_{1to1} in the coalition.

5.5.1.2 Hypotheses

The following hypotheses are the predicted outcome of the experiments:

*Hypothesis H*5.1: Increasing the number of agents n will increase the coalition value $\sum_{j=1}^{n} v_j$.

The number of agents n is the main element and predicted to slow down the process of evaluating dependence relationship. The agents' number will greatly increase chances of encountering invalid relationships. It is also believed that agents' total weight will easily surpass agent agt_{root}'s maximum capacity. Increasing number of agents will increase the coalition value $\sum_{j=1}^{n} v_j$ as well. Thus, a larger coalition formed denotes a higher coalition weight and coalition value is generated.

*Hypothesis H*5.2: The communication rate μ in KDRVM will have an exponential growth rate corresponding to the increase in agents' number n.

The communication rate μ denotes the complete interaction between agents during the process of KDRVM. The communication growth rate is predicted to be

exponential. This phenomenon is suggested in the Metcalfe's law and communication between agents in a coalition is predicted to be n^2. Hence, a large coalition is projected to have a more intense communication rate compare to smaller one.

*Hypothesis H*5.3: The coalition with dep_{or} relationship is predicted to have a least weight compared to other types of dependence relationships.

The dep_{or} relationships provide the agent agt_{root} with options of depending on any agents $agt_j \in Agt_j$ to achieve the common goal. The flexibility of choosing child agents allows agent agt_{prt} to depend on the agent with the least weight. Compared to dep_{or} relationship, the dep_{and} relationships have to depend on every child agents to validate the dependence relationship. However, we believe that this does not guarantee the dep_{1to1} relationship will always have the least weight compared to other relationships. By concluding all these conditions, the dep_{or} relationships are believed to obtain the least weight compared to the other two types of relationships.

5.5.2 Experiment 5.1: Number of Agents

In the first experiment, the impact of increment of agents' number on coalition value and communication rate is studied. The simulation of Experiment 5.1 has the following parameters:

$$[\varphi_{root} = 10000, \sigma_{\min} = 1, \sigma_{\max} = 100, P_{and} = 0.33,$$

$$P_{or} = 0.33, P_{1to1} = 0.34, v_{\min} = 1, v_{\max} = 100]$$

Figure 5.8 and Figure 5.9 show the results of the simulation:

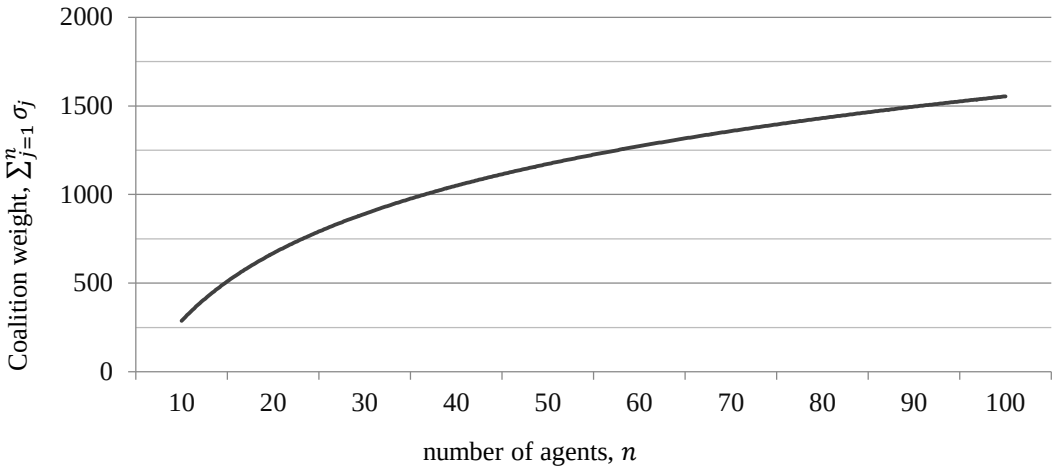


Figure 5.8 Coalition total weight when increasing number of agents

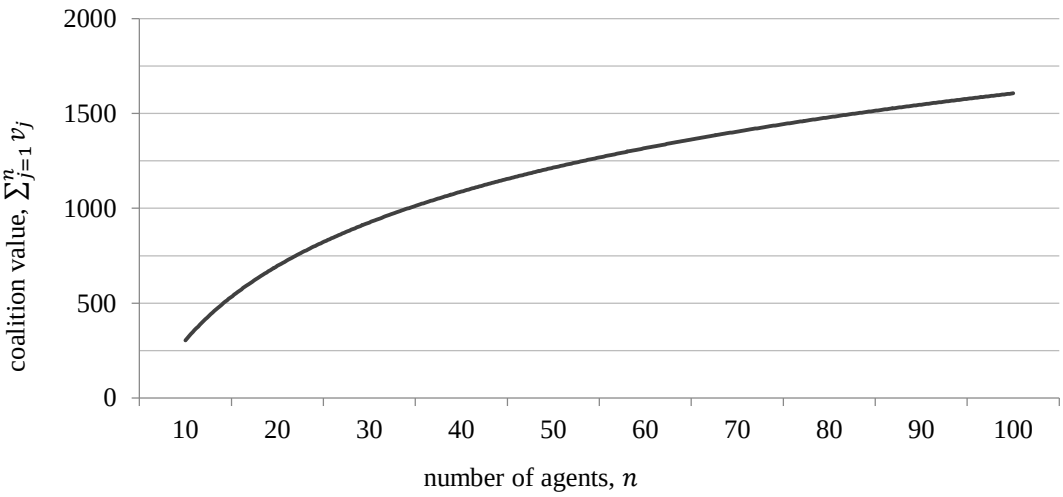


Figure 5.9 The coalition value corresponding to the number of agents

Figure 5.8 shows number of agents has increased the coalition weight from 338 until 1504. A significant growth is observed when n increased from 10 to 30. Once the number of agents reaches 40, the growth of the coalition weight begins to slow down. It is because the coalition total weight is bounded by the maximum capacity. The main reason is the dependence relationship has become complicated as number of agents increases. The number of relationships inside a coalition can be exponentially increased according to number of agents and become complicated.

Hence, the growth rate of the coalition weight at the later stage turns out to be slower.

In Figure 5.9, number of agents has increased the coalition value from 345 until 1560 which is an increase of 451%. This shows coalition value increased corresponding to increment of agents' number. However, the coalition value is indirectly bound by total coalition weight and agent's number will increase the total coalition weight as well. Hence, the Hypothesis H5.1 is accepted as the coalition value has increased along with the number of agents.

The communication rate μ is essential for determining number of interaction occurs between agents during the KDRVM process. Figure 5.10 shows the change in communication rate due to changes in number of agents.

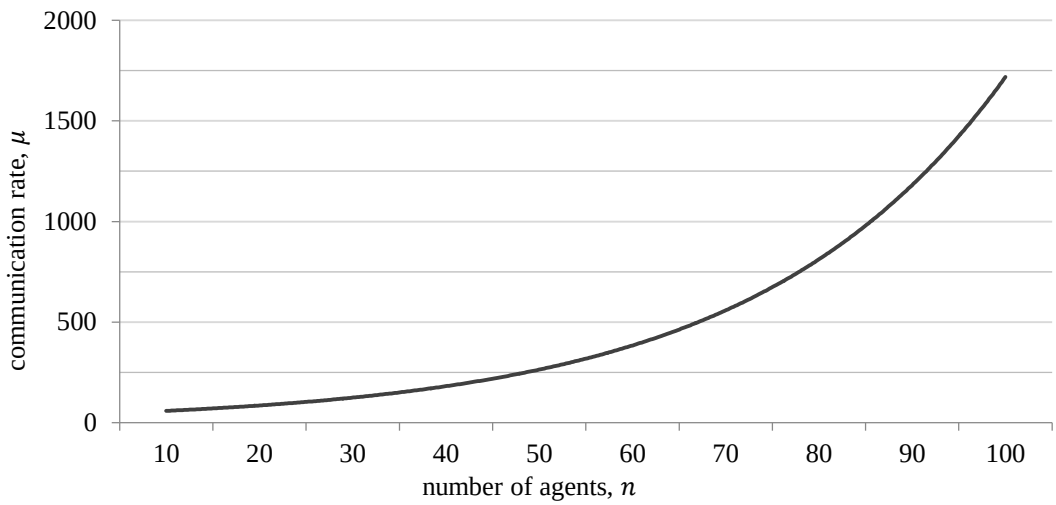


Figure 5.10 Agents' number corresponded to the communication rate inside a coalition

From the Figure 5.10, the growth rate of the communication rate shows an exponential increase over the incremental of agents' number n. Starting at number

of agents n=10, communication rate between agents has a record of $\mu_{n=10} = 30$. When the agents' number reaches n=100, the communication rate has escalated to $\mu_{n=100} = 1143$. Hence, the interaction between agents has shown a significant growth rate. However, the interaction between agents is not proportional to the number of agents by power of two as suggested by Metcalfe's law. Thus, the Metcalfe's law does not apply in this phenomenon.

5.5.3 Experiment 5.2: Type of Dependence

In the second experiment, the types of the dependence relationships correspond to coalition value and communication rate are studied. The simulation of the experiment two has the following parameters:

$$[n = 20, \varphi_{root} = 10000, \sigma_{\min} = 1, \sigma_{\max} = 100, v_{\min} = 1, v_{\max} = 100]$$

In this experiment, there are three types of dependence relationship studied which are dep_{and}, dep_{or} and dep_{1to1}. The ratio P_{and}=1.0 denotes coalition only consists of dep_{and} relationships where the ratio P_{or}=1.0 denotes the dependence relationship in the coalition only compose of dep_{or} relationships. The ratio P_{1to1}=1.0 shows every dependence relations in the coalition are dep_{1to1} relationships. The mixed relationship P_{mix} shows every relationship of a coalition is in a balance state which are P_{mix} = { $P_{and} = 0.33, P_{or} = 0.33, P_{1to1} = 0.34$}. Figure 5.11 and Figure 5.12 show the simulation result of different dependence relationships and its impact on coalition weight and coalition value.

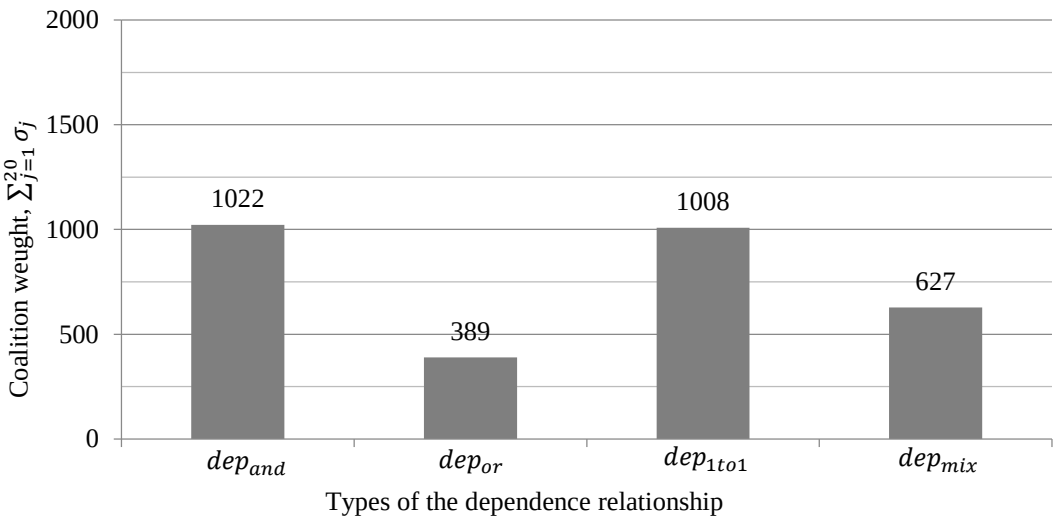


Figure 5.11 Types of dependence relationships corresponded to coalition weight

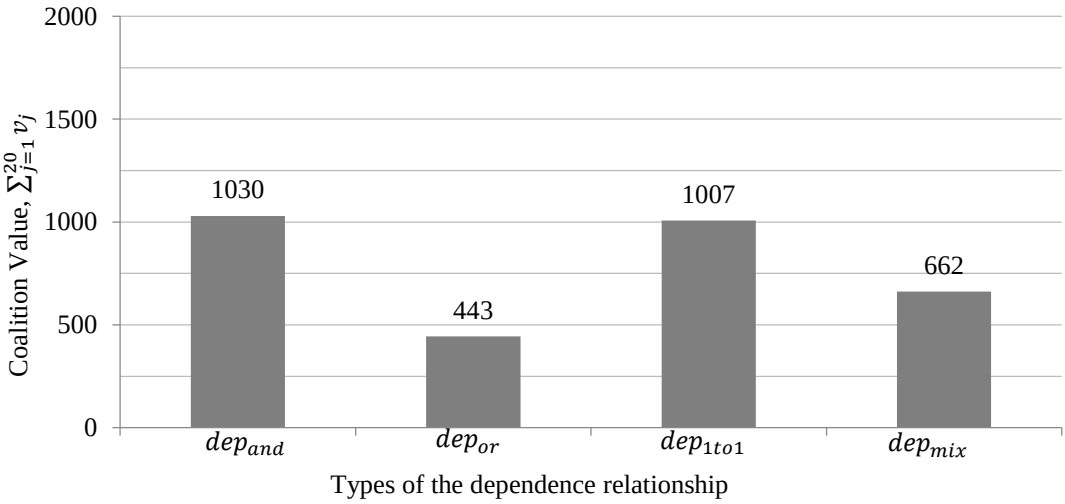


Figure 5.12 Types of dependence relationships corresponded to coalition value

Based on Figure 5.11, it shows P_{and}=1.0 has the highest weight $\sum_{j=1}^{20} \sigma_j dep_{and}$=1022 compared to other types of dependence relationship. Followed by P_{1to1}, which has the total weight of $\sum_{j=1}^{20} \sigma_j dep_{1to1}$=1008. The P_{or}=1.0 and $P_{mix} = \{0.33, 0.33, 0.34\}$ have the total weight of $\sum_{j=1}^{20} \sigma_j dep_{or}$=389 and $\sum_{j=1}^{20} \sigma_j dep_{mix}$=627. The main reasons of a lower coalition weight in the P_{or} and P_{mix} are the dependence relationships provide options

for choosing child agent to cooperate. During dep_{or} relationship, agent agt_{prt} does not have to include all child agents in the relationships for verification. Hence, the total weight of dep_{or} relationship is the lowest compare to other types of dependence relationships.

From Figure 5.12, the P_{and}=1.0 has the highest coalition value which is $\sum_{j=1}^{20} v_j dep_{and}$ =1030. The second highest coalition value is P_{1to1} and the coalition value is $\sum_{j=1}^{20} v_j dep_{1to1}$ =1007. Other types of dependence relationship have recorded the coalition value of $\sum_{j=1}^{20} v_j dep_{mix}$ =662 and $\sum_{j=1}^{20} v_j dep_{or}$ =443 for P_{mix} and P_{or} respectively. The coalition value of the ratio P_{or}=1.0 is significantly lower than others but agent agt_{root} is able to achieve the goal g_{root} with a lesser number of agents. The P_{or}=1.0 is efficient in achieving goals but it does not have the highest profit compared to others. Hence, Hypothesis H5.3 is accepted based on the observation from Figure 5.12 where the P_{or} has the least coalition weight and value.

The rate of communication between agents corresponds to different types of dependence relationships are studied as well. Figure 5.13 shows result of communication rate between agents in the simulation.

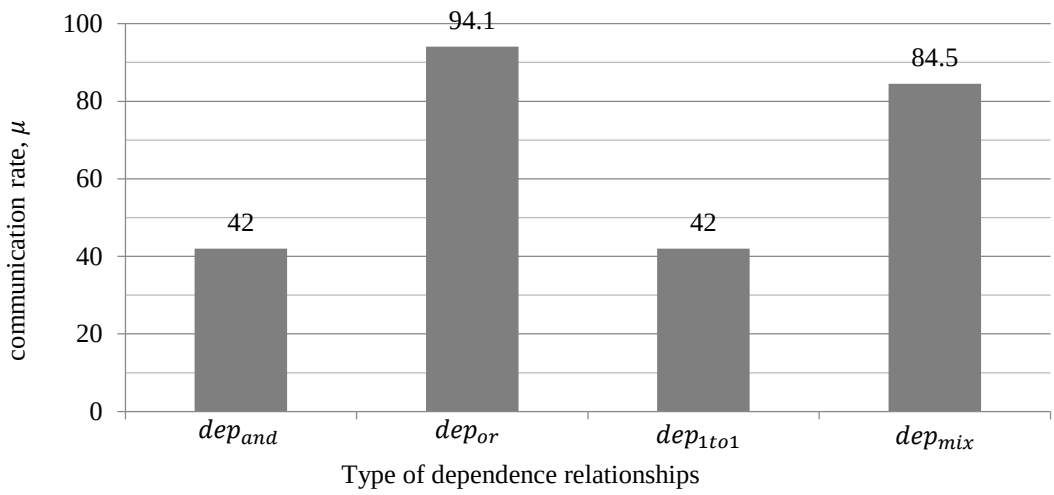


Figure 5.13 Communication rate corresponding to types of dependence relationships

According to Figure 5.13, the P_{or} has the most intense communication rate which is μ_{or}=94 compare to others. The P_{mix} has the communication rate of μ_{mix}=84 while the P_{and} and P_{1to1} shares the same communication rate $\mu_{1to1} = \mu_{and}$=42. The communication rate in $P_{or} = 1.0$ coalition is highest because agent agt_{root} needs to evaluate every child agents in relationship. If one of the child agents Agt_j's weight exceeds agent agt_{root}'s maximum capacity, the dep_{and} relationship is not valid. The interaction between parent and child agents will be halted if the terminate signal is given. However, if $agt_j \in Agt_j$ has a lower weight than agent agt_{root}'s capacity, it needs to traverse through its child agents. This results in the situation with the dep_{or} relationsahip where communication between agents will be intensified.

5.6 SUMMARY

In the Experiment 5.1, Hypothesis H5.1 is accepted as the number of agents gradually increases coalition weight and value. On the other hand, Hypothesis H5.2 is also accepted since the communication rate between agents increased in

exponential but Metcalfe's law does not apply. Hypothesis H5.3 based on both Experiment 5.1 and 5.2 are accepted as well. The dep_{or} relationship has shown a lower coalition weight and value compared to other types of dependence relationships. It allows the coalition to form at a lower cost but also with a lower profit. However, the drawback of the dep_{or} relationship is communication rate is slightly higher compare to the dep_{and} and dep_{1to1} relationships in average condition.

Chapter 6 Join Coalition Mechanism for Macroscopic Coalition

6.1 INTRODUCTION

In this chapter, the JCM is proposed to help an *idle agent* join a macroscopic coalition. It is important to help an idle agent to increase their utility as this will lead to increment of overall society welfare. The JCM is based on the two phase evaluation process which includes evaluation of joining agent's JCR and trial joining process through prisoner dilemma (PD) games.

Prisoner dilemma (PD) is one of the well-known non-cooperative game theory that exhibits an agent properties (Robert Axelrod, 2006) with partial knowledge. The PD game has been introduced by Flood and Desher in the early 1950 and studies *decision making* between two players without advanced knowledge of each other strategy. The basic model of the PD game is explained in the following Example 6.1 (Osborne & Rubinstein, 1994).

Example 6.1: Two prisoners have been arrested and isolated from one another in different cells. They will need to decide whether to cooperate or defect during the interrogation. They have been given offer by an officer that if both prisoners decide to cooperate, they will receive a payout of cooperate payout, C. If the agents decide to defect, they will receive defect payout, D. If one of the prisoners decides to

cooperate while the other prisoner defects, the cooperative prisoner will receive sucker payout, S. Otherwise, the prisoner who defect the cooperative prisoner will receive the temptation payout, E.

By combining the score and prisoners' move from Example 6.1, the C, D, E and S can be represented with the values of $C = 3, D = 1, E = 7$ and $S = 0$ for constructing the basic payout of the PD game. It can be formalized into a general form such as following: $E > C > D > S$ and $C < \left(\frac{S+E}{2}\right)$ which holds the values of payout.

The PD games with n number of players and played in an iterated manner is known as n-players iterated prisoner dilemma (NIPD) (Robert Axelrod, 2006) game. It shares some common characteristics with MAS problem such as (1) number of agents, (2) limited information, (3) decision making, (4) noise of the environment and (5) interaction between agents. The interaction experience is a core element during decision making, an agent needs to consider the past experience and evauate the risk of cooperating with the same agent. The first PD games competition was held by Axelrod (R. Axelrod, 1980) and the strategy won was the Tit for Tat (TFT) submitted by professor A. Rappoport. The TFT strategy proposed an agent attempt to cooperate in the first move and subsequently copy opponent's move from the previous interaction. The TFT strategy performed well in a noisy environment despite its behaviour of being easily provoked. The provocation indicates an agent will defect if it met the same agent which previously choose the defect strategy. The noise in the NIPD games is the agents' response to the opponent which has been manipulated to a certain degree. The result of NIPD games come out different than

the agent intended. Agents made incorrect decisions in a PD games could end up losing games and profit. To improve the behaviour of making wrong decision, Rogers, Dash, Ramchurn, Vytelingum, and Jennings (2007) proposed an extension to help players to resolve their mistakes and earn more points. It uses pre-arranged sequence of moves and a round-robin style approach for agents take turns to win the game. In addition, Choi (2007) has pointed out the effect of "trembles" in a NIPD game which can be used to improve the cooperation between agents. The trembles show the agent has won NIPD games and further increases the motivation of continued cooperation with opponents. Besides, L. G. Nardin and Sichman (2010); Luis Gustavo Nardin and Sichman (2011) used the spatial PD game to study mutual trust issue between agents during CF.

According to Jonker and Treur (1999), trust between agents is defined as the attitude of an agent with respect to the dependability or capabilities corresponding to the flow of the events. There are two well-known classification trust models which are Sabater and Sierra (2005)'s and Balke, Konig, and Eymann (2009)'s classification. The Sabater et al.'s classification dimension covers major elements of the trust aspects while maintaining a general view of the reputation and characteristics of the trust mechanism. On the other hand, the Balke et al.'s classification focuses on five stages of transition which are (1) recording cooperative behaviour, (2) rating cooperative behaviour, (3) storing the cooperative behaviour, (4) recall the cooperative behaviour and (5) adapt or learn the cooperative behaviour. The majority of trust models presented emphasizes on direct interaction between agents. Direct interaction implies the communication and experience of an agent to revise its belief without referring to external information. According to Ramchurn, Sierra, Godó, and

Jennings (2003), the trust of an agent within a BDI architecture is divided into *confidence* and *reputation*. An agent's confidence is built on top of the interaction experiences with other agents and has a direct impact to its belief. An agent's reputation is gained through the information exchange with other agents in the MAS society. The reputation of an agent is important when there is no past interaction history between agents. On the other hand, a further derivation of agent's confidence has been presented by Huynh et al. (Huynh, Jennings, & Shadbolt, 2006) and is categorized into (1) interaction trust, (2) role-base trust, (3) witness reputation and (4) certified reputation. Many well-known trust models such as ReGreT systems (Sabater & Sierra, 2001), AFRAS (Carbo, Molina, & Davila, 2003), FIRE (Huynh et al., 2006), Marsh (Marsh, 1994) and LIAR (Muller & Vercouter, 2005) are based on the direct interaction while considering other elements for revising an agent's belief. The detailed survey of other trust models have been reviewed by Sabater and Sierra (2005). The trust metric used in the JCM is similar to the Marsh's model (Marsh, 1994). Marsh has proposed the formalism of trust between agents using the value of 0 and 1 which is $T_\alpha \in \{0,1\}$. These numbers are purely comparative values and do not represent any meaning at all. When the trust value of an agent is 0, it represents a complete distrust towards other agents. Otherwise, if the trust value is 1 represents a complete blind trust between agents. Similar formalism has been found in Gambetta's work (Gambetta, 1990).

The motivation of this chapter is to helps idle agents to increase their utilism by joining an existing macroscopic coalition. The main reason that we have used the non-cooperative game theory approach is to simulate the properties of the partial knowledge property that an agent possesses. The trial joining process advocates a

similar concept and that is why we have chosen it as the interaction between the joining agent and the targeted coalition. In our proposed model, agents are assumed to be non-cooperative if they are not engaged in any coalition. However, coalition members will cooperate with each other once they have developed mutual trust through CF.

First, the preliminaries for denoting mathematical notations are presented followed by the approach for designing JCM for macroscopic coalition. The design of JCM includes different perspectives of agents in the coalition. Subsequently, experiments have been carried out, analysed and discussed. Finally, the summary of this chapter is outlined.

6.2 MATHEMATICAL NOTATIONS

In this section, the mathematical notations for JCM are presented. First, agent's behaviour is described followed by the ratio of agents in the coalition. Subsequently, the interaction between agents using the NIPD games is discussed. Lastly, the trust metric is presented.

Definition 6.1: The agent's behaviour Q_{agt} is the particular agents' strategic corresponding to the NIPD games. In this chapter, there are five behaviours involved in the NIPD games. The following Table 6-1 shows agents' strategic, behaviours and the descriptions:

Table 6-1 The agents' strategy in a PD game

Type of Strategies	***Behaviour***	***Descriptions***
cooperate	desperate	Always choose cooperate with others.
defect	grumpy	Always choose not to cooperate with others.
tit for tat	imitator	Cooperate on the first time and mimic the other agents' move
unforgiving	avenger	Cooperate until opponent defect and continuously defect
random	uncertain	Cooperate 50% and defects 50% of the time.

The five strategies from Table 6-1 are the well-known approaches of PD games which can be found in Marko et al.'s paper (Jurisic, Kermek, & Konecki, 2012).

Definition 6.2: The coalition ratio $\sum_{u=1}^{5} P_{su}$ is the agent's ratio with five different behaviours which are (1) "desperate" agents P_{s1}, (2) "grumpy" agents P_{s2}, (3) "imitator" agents P_{s3}, (4) "avenger" agents P_{s4} and (5) "uncertain" agents P_{s5}.

The number of strategies S represents the quantity of possible strategies in a coalition. Each strategy is a disjoint set of combination and complies with third axiom of the probability axioms for denoting the ratio of the coalition. The coalition ratio is denoted as the following equation:

$$\sum_{u=1}^{S} P_{su} \tag{6.1}$$

The total ratio of a coalition is 1.0 and it complies with second axiom of the probability axiom. The coalition society ratio $\sum_{u=1}^{5} P_{su} = P_{s1} + P_{s2} + P_{s3} + P_{s4} + P_{s5}$ where $\{P_{s1}, P_{s2}, P_{s3}, P_{s4}, P_{s5}\}$ represent the ratio of agents with the behaviour $Q_{agt} = \{$ desperate, grumpy, imitator, avenger, uncertain$\}$ respectively. According to the probability axioms, the ratio of each strategic must be greater or equal to 0.

Assumption 6.1: Agent in the coalition z, agent $agt_x \in z$ adopts the sincerity principle when interacting with other agents.

This suggests coalition members $agt_x \in z$ trust each other since they formed collaboration. Also, they will respect each other's strategy in a NIPD game session when received the JCR from agent agt_{join}.

Assumption 6.2: The communication channel between agents is assumed to be noiseless and lossless.

There is no disturbance in the communication channel and no interruption occurs when agent agt_{join} sends the JCR to the targeted coalition through agent agt_{reg}.

Definition 6.3: The iteration number, d of the PD game is a positive odd number.

It is denoted as following:

$$d = \{2l + 1; \forall l \in RE \cap \forall l \geq 0\} \tag{6.2}$$

The game iteration number, d represents number of games played for determining the winner. Since it is using odd number, a winner is guaranteed for each game

session. The l is the positive real number enforces the length of trial joining period that an agent agt_{join} wish to cooperate with the coalition z in the second phase. The second phase is trial joining process where NIPD games are commenced. The higher value l shows more PD games will be played to ensure a long term trust constructed between agent agt_{join} and coalition member $agt_x \in z$.

Definition 6.4: There are two conditions for the agent agt_{join} to win the PD game session.

The agent agt_{join} must either (1) get the agents $agt_x \in z$ in the targeted coalition to cooperate and receive cooperate payout or (2) choose to defect against agents $Agt_x \in z$ and receive the temptation payout. The following Table 6-2 shows the winning conditions for the agent agt_{join} in a PD game:

Table 6-2 Winning condition for agent agt_{join} in a PD game

	$agt_x \in z$ ***cooperate***	$agt_x \in z$ ***defect***
agt_{join} ***cooperate***	3, win	0, lose
agt_{join} ***defect***	5, win	1, lose

The agent agt_{join} need to decide whether to cooperate or defect to win the agent $agt_x \in z$ without prior knowledge about $agt_x \in z$ ’s move.

Definition 6.5: Trust factor T_α represents an agent agt_{join}’s trust metric towards the targeted coalition.

It can be represented as the following:

$$T \in \{0,1,...,10\} \tag{6.3}$$

where the trust factor T_α has minimum value of 0 and the maximum value of 10. Throughout the NIPD games, the number of games won H and lost $\overline{H}$ will be monitored by CRA agt_{reg} to decide the overall winner. The following Table 6-3 shows the changes to the trust factor, T_α according to the PD games' results:

Table 6-3 Trust factor disbursement according to the NIPD games' result

	Agent agt_{join} ***'s trust factor,*** T_α ***towards agent*** $agt_x \in z$
Win majority game, $H \geq \overline{H}$	$T_\alpha + 1$
Lose majority game, $H > \overline{H}$	$T_\alpha - 3$

Definition 6.6: Trust threshold T_ω is the baseline of the trust factor.

In our proposed method, we emphasize on the agent agt_{join} trust factor. We assume the coalition members are cooperative agents and the trust has been built through their past interaction. A healthy trust factor is a trust factor is higher than the trust threshold as shown in the following equation:

$$\sum_{h=1}^{n} T_{\alpha h} < T_\omega \tag{6.4}$$

Our proposed trust model in JCM is similar to Marsh's formalism (Marsh, 1994) and Gambetta's work (Gambetta, 1990). The trust factor in the JCM focuses on agent agt_{join}'s perspective when it proposes a JCR to the coalition. If the trust factor T_α is

below the trust threshold T_{ω}, agent agt_{join} will not join the coalition as agent agt_{join} loses its trust towards the targeted coalition z.

Definition 6.7: The communication rate μ is the percentage of committed interaction out of all possible interaction.

The communication rate μ in this chapter is calculated using the following equation:

$$\mu = 100\% \times \left(\frac{m}{M'} \right) \tag{6.5}$$

where m represents the number of communication section committed and M' represents total possible interaction in JCM. The total interaction between agent agt_{join}, agt_{reg} and $Agt_x \in z$, the total possible interaction is calculated using the following notation:

$$M' = \sum_{h=1}^{n} (2 \times d_h) \tag{6.6}$$

where the n represents number of agents and d represent the PD games sections commenced in Equation 6.2.

6.3 APPROACH

The purpose of the JCM is to help idle agents join an existing macroscopic coalition. There are several factors to be considered when designing the JCM such as spamming and reliability. The JCM starts with an agent agt_{join} that has the intention of joining a macroscopic coalition to maximise its profit. This intention consists of the set of goals G_{join} that an agent agt_{join} wants to achieve. It will try to achieve its

goal through cooperation with others if it is not autonomous of its capabilities (note: not S-Autonomous). Hence, it will seek helps from other agents with the required capabilities. Figure 6.1 shows the flow chart of an agent agt_{join} attempt to join a targeted coalition.

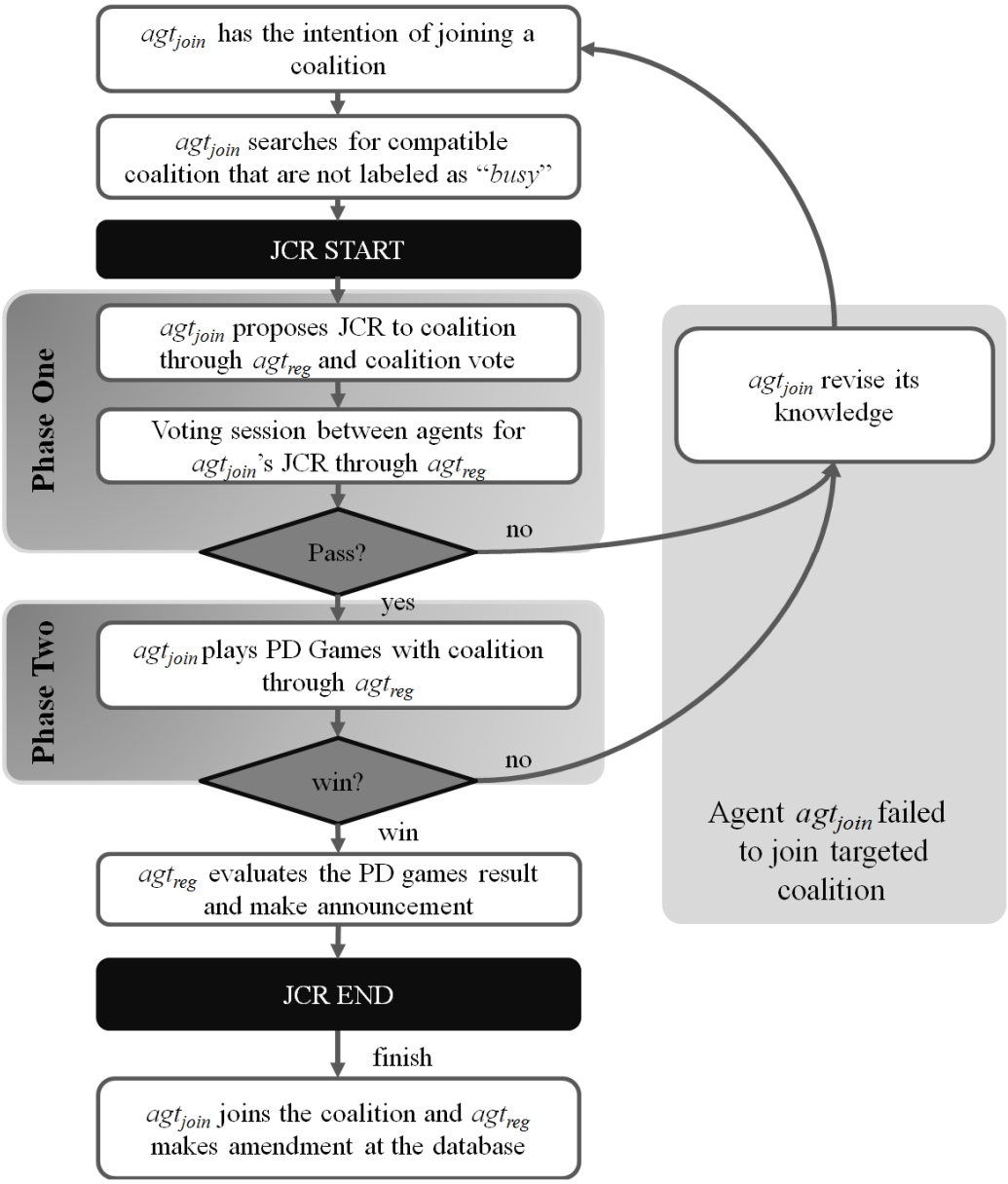


Figure 6.1 The overall procedure of agent agt_{join} attempts to join the coalition.

The JCM consists of two-phase evaluation process which are sequential voting and

NIPD games. The agent agt_{join}'s targeted coalition is chosen from the list of coalition (LOC) generated and sent by agent agt_{reg}. It is composed of the coalition lists that is compatible with the agent agt_{join}'s goal g_{join}. The coalition with the label of "busy" will be filtered out in the LOC when sent to agent agt_{join}. After choosing the coalition with the highest compatibility, the agent agt_{join} will send the JCR to the targeted coalition through agent agt_{reg}. The JCR contains the list of goals G_{join} and the agent agt_{join}'s offer. Subsequently, the coalition member will have to vote for it. The first phase ends with the announcement of voting result. The voting session is commenced between agt_x in the coalition for deciding the agent agt_{join}'s JCR.

The second evaluation phase is the trial joining session with NIPD game session between agent agt_{join} and Agt_x through agent agt_{reg}. The NIPD games session will be monitored by the agent agt_{reg} to prevent any unfair result. In a nutshell, the voting results will determine the initial acceptance of the agent agt_{join} in term of its offer and the second phases help agent agt_{join} to evaluate the trustworthiness of the coalition. The JCM is based on the three perspectives of different agents during the process of agent agt_{join} joining a coalition. These perspectives are divided into agt_{join}, agt_{reg} and $agt_x \in z$ respectively which is derived from the roles of agents in Chapter 2.

6.3.1 Joining Agent's Perspective

The idle agent has the main role of the JCM if it has fulfilled the three condition of the joining agent agt_{join} to propose the JCR (note: the three conditions are shown in

the Chapter 2). Figure 6.2 shows the algorithm for agent agt_{join} in the JCM:

ALGORITHM 6.1. Agent agt_{join} joins a macroscopic based coalition.
1. Searches for a compatible coalition where both have common goals through the agt_{reg}. 2. Retrieves the LOC from the agent agt_{reg} and decides on the targeted coalition. 3. Send the JCR to targeted coalition through agt_{reg}. 4. Waits for the reply from the agent agt_{reg} regarding the voting session result. 5. If the agent agt_{join} did not receive the agt_{reg} within the interval or received acknowledgement of JCR's failure, it will search alternate compatible coalitions by repeating Step 1. Otherwise, the agent agt_{join} will proceed to Step 6. 6. Play NIPD games with the targeted coalition by sending decision(s) to agt_{reg}. 7. If $\sum_{h=0}^{n} T_{\alpha h} < T_{\omega}$, agt_{join} will revise its belief and repeat Step 1. Else, it will wait for agt_{reg} announcement to join the coalition. 8. Officially joins the coalition and wait for tasks scheduling.

Figure 6.2 The algorithm based on agent agt_{join}'s perspective in JCM

The algorithm in Figure 6.2 shows agent agt_{join}'s perspective in the JCM where it will undergo two-phase evaluation process with agent agt_{reg} and targeted coalition z. A voting session will be organized to decide agent agt_{join}'s JCR. The agent agt_{join} will wait for the announcement of the phase one's result. If agent agt_{join}'s JCR is rejected by the targeted coalition, it will propose JCR to other compatible coalitions in the LOC while revising its knowledge. The revision of agent agt_{join}'s knowledge demotes the coalition's credibility. If agent agt_{join}'s JCR evaluation passes, the second evaluation will be conducted using NIPD games. The score of the

PD games will decide the credibility and commitment of agent agt_{join}. The result of the NIPD games will decide the final joining process of agent agt_{join}. The NIPD games serves as a trial interaction between agent agt_{join} and coalition to ensure trustworthy and commitment of each other. It is possible to consider coalition's reputation from neighbors but we implement the trial joining process to obtain the agent's credibility through direct interaction. If the agent agt_{join} trusts the targeted coalition after the trial joining phase, it will continue to cooperate with coalition. Otherwise, it will search for an alternate coalition if available.

6.3.2 Coalition Registration Agent's Perspective

The second algorithm is agent agt_{reg}'s perspective who serves as the middleman agent between agt_{join} and $Agt_x \in z$ during JCR agt_{join}'s Proposal. Figure 6.3 shows the algorithm and functions of agent agt_{reg} in the JCM.

ALGORITHM 6.2. Agent agt_{reg} handling the agent agt_{join}'s JCR.
1. Receives request of joining coalition from the agt_{join} with the goal g_{join}. 2. Evaluates the agent agt_{join} and ensures it is not engaged in any recent JCR activities (prevents spamming). 3. Searches the lists of compatible coalition with the goal g_{join} and return LOC to agent agt_{join}. 4. Receives JCR from agent agt_{join} and the targeted coalition. 5. The coalition z will be labelled "busy" for certain interval to prevent spamming. (See next page) 6. Organizes the voting section among the coalition members.

7. If the voting section for the JCR passes, acknowledges agent agt_{join} and $agt_x \in z$ before proceeds to the Step 8. Otherwise, acknowledge agent agt_{join} to terminate its JCR proposal.
8. Conducts NIPD games monitoring between agent agt_{join} and coalition members $Agt_x \in z$ through agent agt_{reg}.
9. Records the NIPD game results in H and $\overline{H}$. If $H < \overline{H}$, agent agt_{join}'s $T_\alpha - 3$. If $H \geq \overline{H}$, agent agt_{join}'s $T_\alpha + 1$.
10. Announces the PD games result to agent agt_{join} and coalition.
11. Updates the database if agent agt_{join} managed to win majority game.

Figure 6.3 The algorithm based on agent agt_{reg}'s perspective in JCM.

The agent agt_{reg} is not permitted to engage in any coalition activities because it has the role as a judge for voting session and NIPD game sessions. In other words, agent agt_{reg} has to be neutral to ensure the fairness of the voting and trial joining result. Despite that, agent agt_{reg} will gain incentives from agent agt_{join}'s offer in the JCR if agent agt_{join} successfully join the targeted coalition. By successfully help agent agt_{join} join a coalition, agent agt_{reg} is able to maximize its profit as a middle man agent. The agent agt_{reg} gains incentives from agent agt_{join} by helping the agent agt_{join} to join the targeted coalition. The agent agt_{reg} has the access to database where information on the coalitions of MAS society are stored. Throughout the agent agt_{join}'s JCR, agent agt_{reg} will access the database to generate the LOC for the agent agt_{join} with compatible set of goals. Also, if agent agt_{join} successfully join the coalition, the agent agt_{reg} is reponsible to amend the database for the updated coalition.

The calculation of votes will be stopped if it passes the quota of 50% of the n agents inside the coalition. The agent agt_{reg} will monitor the agent agt_{join}'s trust and determine the NIPD games result. The trust model in the JCM implements a "slow gain" approach where agent agt_{join}'s trust factor is awarded $T_{\alpha}+1$ for any winning game in the NIPD game session. However, the losing of PD games penalises agent agt_{join}'s trust factor $T_{\alpha}-3$. If the trust factor is lower than the designated threshold, the agent agt_{reg} will inform agent agt_{join} that it failed in the second phase. Otherwise, agent agt_{reg} will inform agent agt_{join} to get ready to join the targeted coalition.

6.3.3 Coalition Member's Perspective

The third algorithm is the perspective of coalition members $Agt_x \in z$ in the coalition during the agent agt_{join}'s proposal of JCR. Based on Assumption 6.3, coalition members trust and respect each other for the voting result towards agent agt_{join}'s JCR. This also applied to the decision of coalition members when choosing strategies during NIPD games session. Figure 6.4 shows the algorithm for agent $agt_x \in z$ in the JCM.

ALGORITHM 6.3. Agent $agt_x \in z$ handling the agent agt_{join}'s JCR.
1. Receives notification from agent agt_{reg} about agent agt_{join}'s JCR. 2. Evaluates the agt_{join}'s JCR. 3. Votes by sending the decision to agent agt_{reg}. (See next page) 4. Waits for agt_{reg}'s acknowledgement of the voting result.

5. If majority votes agree on agent agt_{join}'s JCR, proceed to Step 6. Otherwise, terminate agt_{join}'s JCR and continue tasks.
6. Sends decision to agent agt_{reg} to play NIPD games with agent agt_{join}.
7. Waits for agent agt_{reg}'s announcement of the NIPD game result.
8. If NIPD games show agent agt_{join} won, wait for coalition reformation. Else, the coalition agents will continue with their current tasks.

Figure 6.4 The algorithm for agent $agt_x \in z$'s perspective in JCM.

The JCM algorithm for $agt_x \in z$ begins with the JCR notifications from agent agt_{reg} and interrupts their current tasks. The agents Agt_x are required to vote for agent agt_{join}'s JCR based on the evaluation of offer and coalition goals. After the voting session, the $Agt_x \in z$ will wait for agent agt_{reg}'s announcement on the voting result. If agent agt_{join} managed to pass the voting section, the NIPD games session will commence for agent agt_{join}'s trial joining. Each strategies denote agent agt_x's reaction towards agt_{join}'s JCR. The "desperate" and "imitator" are categorized as cooperative approaches in this chapter. If the agent agt_{join} won majority of the NIPD games, agent agt_{reg} will initialize a coalition reformation order and the total number of members will be increased by one. The reformation of coalition includes sending updated members list to agent agt_{reg} and reschedule plan or tasks by including agent agt_{join}.

6.3.4 Computational Complexity Analysis

The computational complexity of JCM algorithms are divided into three sections according to different agents' perspective. The following are the analysis of the computational complexity of each algorithm. The computational complexity for the

Algorithm 6.1 in Figure 6.2 is O($\Re + (n \times d)$) where $\Re$ represents the number of coalitions in LOC for the agent agt_{join} to search. The $n \times d$ represents the total session of NIPD games played. The LOC generated by agent agt_{reg} is in sorted order according to the percentage of goal compatibility. The worst case for the Algorithm 6.1 is when agent agt_{join} failed to join every available coalition in the LOC and played NIPD games with every coalition member. The Algorithm 6.2 in Figure 6.3 has the complexity of O($d + n + \Im \log(\Im)$) where n represents the number of agents in the coalition and $\Im$ represents the number of coalition registered in database. The computational complexity of searching LOC with the compatible set of goals is O(1) because sorted linked lists are implemented to store the coalition list. The sorting algorithm uses heap sort and computational complexity is O($\Im \log(\Im)$).The voting section uses a sequential voting approach which has a computational complexity of O(n). The NIPD games has a computational complexity of O($2(n \times d)$) where agent agt_{reg} is the middleman agent for conducting NIPD games. The update of the coalition lists use a computational complexity of O($\Im$) if the agent agt_{join} joins the targeted coalition. The third algorithm in Figure 6.4 has a computational complexity of O(NP) and coalition reformation is a NP hard problem. The agents in the coalition have a lesser computational complexity as it only has to evaluate, vote and play NIPD games. The worst computational complexity is the reformation of the coalition and it is a NP-hard problem. Most of the activities for reformation of the coalition are identical to CF and are shown to be *NP-hard* problems (Sandhlom & Lesser, 1997; Sandholm et al., 1999). In a nutshell, the algorithms of JCM are not under category of *NP-hard* problem except for the part of coalition reformation. Further analysis of the algorithms can be found in Appendix E.

6.4 EXPERIMENTAL RESULTS

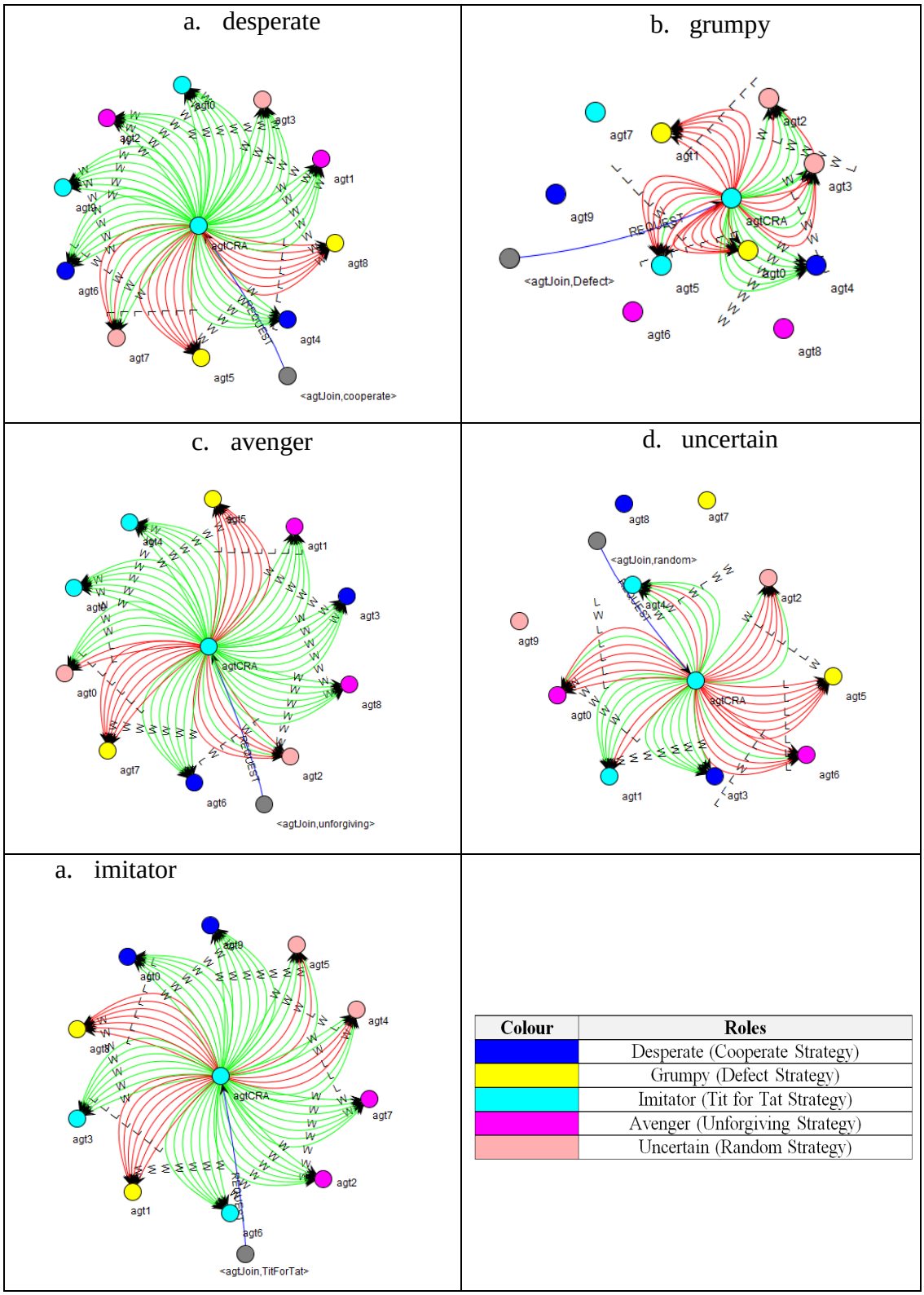
The performances of the experiments are measured using the criteria defined in the following Table 6-4:

Table 6-4 Performance Measurements of the experiments

Performance measurements	***Descriptions***
M'	The possible communication between agent agt_{join} and $agt_x \in z$ in JCM.
m	The communication session established when sending the JCR.
T_{sim}	The simulation time of the JCM (Recorded in nano-second and scaled to second).
$sc = \sum_{x=1}^{n \times d} \{C, D, E, S\}_x$	The total score of the NIPD games played in the iteration by agent agt_{join}
$\aleph = \frac{sc}{(n \times d)}$	Average score of each NIPD games
E_{μ}	The efficiency of the unutilized communication. It is calculated using $100\% - \mu$.
H	The agent agt_{join} game winning counters.
μ	The efficiency of the communication rate sent between agents during JCM.

The simulation of JCM is conducted on the workstation with the following specification: *Dell OptiPlex* 390, *Intel Core i*3 *processor* and 4*GB RAM*. It uses Java Universal Network Graph (JUNG) library (O'Madadhain, Fisher, White, & Boey, 2003) for visualizing the agent agt_{join} joining progress. Figure 6.5 shows the example of agent agt_{join} with different behaviour, Q_{join} is trying to join the targeted

coalition using JCM.



Colour	Roles
	Desperate (Cooperate Strategy)
	Grumpy (Defect Strategy)
	Imitator (Tit for Tat Strategy)
	Avenger (Unforgiving Strategy)
	Uncertain (Random Strategy)

Figure 6.5 The joining process corresponding to the agent agt_{join}'s behaviour

Although the "Defect" strategy is considered a Nash Equilibrium (Osborne & Rubinstein, 1994), but it does not encourage collaboration between agents as both entities does not gain benefit by doing so. The "uncertain" behaviour of agent agt_{join} shows a random approach in choosing a move which it does not guarantee the result of the NIPD game. Also, this strategy will be considered as risky for agent agt_{join}.

Each experiment is conducted with the iteration of 1000 executions. The next subsection is Experiment 6.1 studies the impact of coalition members towrads different ratio of "Grumpy" behaviour agents, P_{s2} during the agent agt_{join}'s JCR. The Experiment 6.2 simulates the different behaviour of agent agt_{join} when proposing JCR and the trial joining request.

6.4.1 Experiment 6.1: Grumpy Agents' Ratio

In this experiment, the impact of varying the "Grumpy" behaviour agents' ratio, P_{S2} on to the rate of communication is simulated. The following hypothesis H6.1 is predicted for the first experiment's result.

*Hypothesis H*6.1: The agent agt_{join} and the targeted coalition is projected to have a lower communication rate in a high ratio "Grumpy" behaviour coalition z. The JCM will terminate if agent agt_{join} failed to join the targeted coalition. Otherwise, if agent agt_{join} is able to maintain its trust factor until last agent agt_n in the targeted coalition,the targeted coalition will undergo a reformation process.

The parameters of the Experiment 6.1 are shown as following:

$$[Q_{join} = \text{“desperate”}, n = 25, d = 7, T_{\alpha} = 10, T_{\omega} = 2]$$

The agent agt_{join}'s strategy in this experiment is “Desperate” and it tries to cooperate with coalition member $agt_x \in z$ during the NIPD game sessions. Throughout the simulation, the population ratio of the “grumpy” behaviour coalition member, P_{s2} in the coalition will increase from 0.0 until 1.0. Figure 6.6 shows the result of communication rate corresponding to different variant ratio of “Grumpy” behaviour coalition member:

The result from Figure 6.6 shows a significant decrease in the communication rate when the ratio of “grumpy” behaviour agents, P_{s2} increases. This phenomenon occurs when agent agt_{join} terminates the JCR for losing its trust towards the targeted coalition. Thus, both agent agt_{join} and the coalition members have stopped communicate with each other through the agent agt_{reg}. On the other hand, the high ratio of the “grumpy” behaviour coalition member P_{s2} tends to defect in the NIPD games and causes the agent agt_{join} to lose its trust. This causes agent agt_{join} to have a lower chance to join the targeted coalition. When the “Grumpy” agents ratio reaches P_{s2}=1.0, agent agt_{join} is able conserved up to 88.0% of communication rate during the JCR proposal.

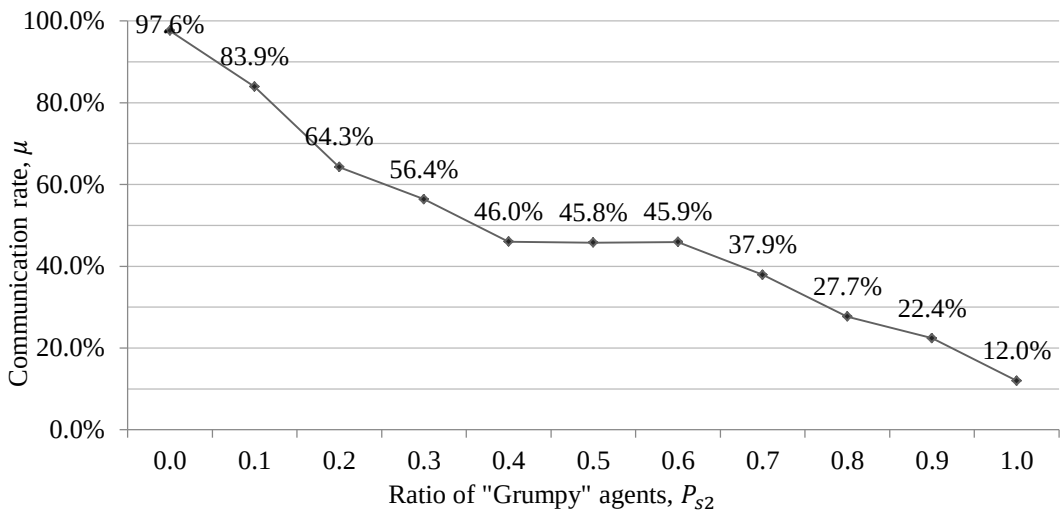


Figure 6.6 The communication rate μ corresponding to the increment of P_{s2}

In the Experiment 6.1, the secondary information are stored during the calculation of scores earned are (1) winning games counter, (2) losing games counter and (3) total scores of the NIPD games.

Table 6-5 The winning, losing counter and the total scores of NIPD games in Experiment 6.1

P_{s2}	H	$\overline{H}$	sc
0.00	959	41	460407
0.10	689	311	344346
0.20	183	817	226305
0.30	20	980	186987
0.40	0	1000	141423
0.50	0	1000	140580
0.60	0	1000	142317
0.70	0	1000	107451
0.80	0	1000	62280
0.90	0	1000	41355
1.00	0	1000	0

The Table 6-5 shows the winning count of NIPD games gradually decreases when the P_{s2} is at 0.4. It shows agent agt_{join} lost in the majority NIPD games and total NIPD game scores decreased from 460407 to 0 when the P_{s2} approaches 1.0. This is because number of coalition members rejecting the agent agt_{join} increases when the P_{s2} increases.

The computational time, T_{sim} for agent agt_{join} to join the targeted coalition has been measured to determine the efficiency of JCM. Figure 6.7 shows the computational time, T_{sim} (in seconds) corresponding to ratio of grumpy behaviour agents in coalition, P_{s2}:

Based on the Figure 6.7's observation, the computational time shares a similar pattern with the communication rate, μ which is decreasing in a linear rate. The main cause of this phenomenon is agent agt_{join} has a higher tendency of not being able to join the targeted coalition and thus terminate its JCR.

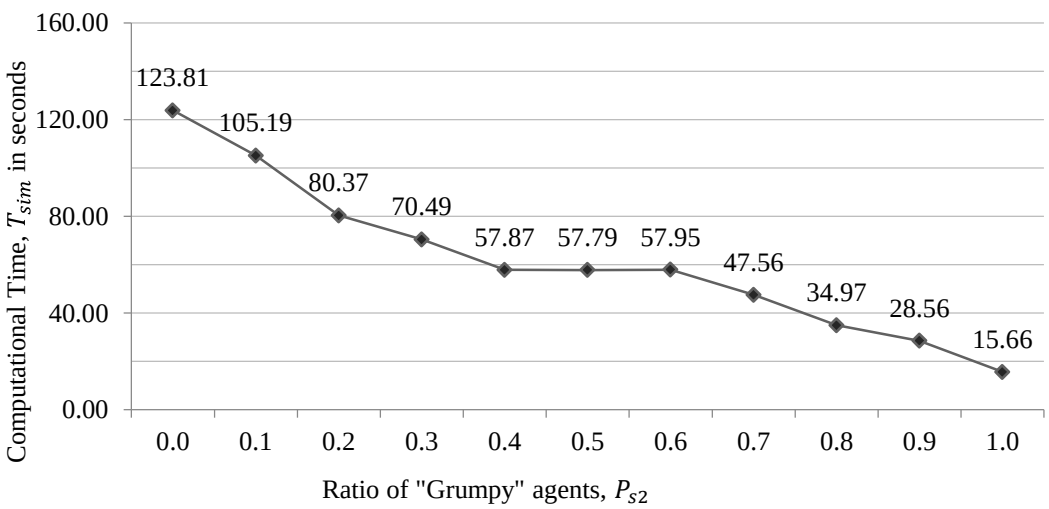


Figure 6.7 The computational time T_{sim} as P_{s2} increases

In Experiment 6.1, the validity of hypothesis H6.1 is accepted where JCM performs better in a high ratio of "Grumpy" behaviour agents, P_{s2} in term of conserving communication rate and computational time. However, the increase of "Grumpy" behaviour agents in the coalition will reduce the chances of agent agt_{join} joining the targeted coalition. It is because agent agt_{join} are not able to win NIPD games with the defective move from coalition member Agt_x. However, the computational time is reduced through the increment of P_{s2}.

6.4.2 Experiment 6.2: The Joining Agent's Strategy

The simulation of agent agt_{join}'s behaviour corresponding to performance measurements are studied in the Experiment 6.2. The "Tit for Tat" strategy performs better among other strategies as shown in the paper (Robert Axelrod, 2006). With this concept in mind, the cooperative behaviour of joining agent agt_{join} is believed to have a higher score in NIPD games. The following Hypothesis H6.2 is the prediction for the second experiment outcome.

*Hypothesis H*6.2: The agent agt_{join}'s with "desperate" behaviour will earn more scores than the "grumpy" behaviour in NIPD games. The agent agt_{join}'s behaviour is projected to have significant influence on performance measurements. There are two conditions of winning the NIPD games according to Table 6-2 which are (1) cooperation between agent agt_{join} and agent $agt_x \in z$ as well as (2) agent agt_{join} has successfully defecting agent $agt_x \in z$. With the cooperative behaviour, agent agt_{join}

will help others as long as there is mutual profit among both parties. Hence, the agents with the "desperate" behaviour are projected to perform better in earning profit through NIPD games.

The following notations are the parameters of this experiment:

$$[n = 25, d = 7, T_{\alpha} = 10, T_{\omega} = 2]$$

where coalition's ratio is in a balance state (Refer to previous chapter for the balance ratio coalition ratio).

Figure 6.8 shows the communication rate, μ according to agent agt_{join}'s behaviour, Q_{join}.

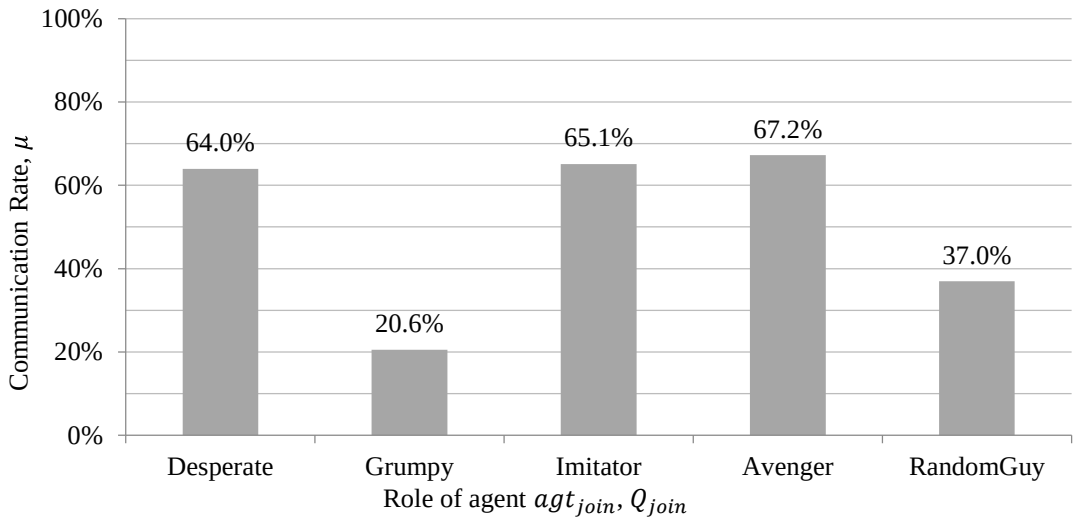


Figure 6.8 The communication rate, μ corresponding to agent agt_{join}'s behaviour

The Figure 6.8 shows Q_{join} = "Grumpy" and Q_{join} = "Random" have a lower communication rate μ compare with other behaviours. The main reason is agent

agt_{join} has a high tendency not to join the targeted coalition and terminates the coalition due to defective behaviour. However, a lower communication efficiency does not imply that agent agt_{join} has a lesser chance of joining the targeted coalition. Mainly because agent agt_{join} will spend more time playing the NIPD games with every coalition member without losing its trust.

Table 6-6 The winning, losing counter and average scores of the NIPD games in Experiment 6.2

Q_{join}	***Desperate***	***Grumpy***	***Imitator***	***Avenger***	***Random***
H	178	0	189	220	3
$\overline{H}$	822	1000	811	780	997
$\aleph$	2.0044	2.4424	2.3365	2.3491	2.3241

Table 6-6 shows the agent agt_{join} with Q_{join}="Grumpy" and "Random" behaviour scores the lowest winning count compare to others. It is because the coalition members are able to counter agent agt_{join}'s "Avenger" and "Imitator" behaviour. Despite agent agt_{join}'s behaviour of Q_{join}="Desperate" having a higher winning counts, it scores the lowest average PD game points. The main reason is agent agt_{join}'s behaviour Q_{join}="Desperate" does not counter the defecting move from coalition members and always chooses to cooperate. In conclusion, the agent agt_{join} with "imitator" and "avenger" behaviours performs better in winning the majority of NIPD games while earning higher scores.

Figure 6.9 shows the computational time T_{sim} corresponding to the agent agt_{join}'s

different behaviour Q_{join} in JCM. The agent agt_{join}'s with the behaviours Q_{join}={ "Desperate", "Imitator", "Avenger"} have a higher computational time, T_{sim} and has a higher chance for agent agt_{join} to join the targeted coalition. The high computational time indicates the agent agt_{join} either play the NIPD games until the last one and got rejected or accepted as a part of the coalition z .

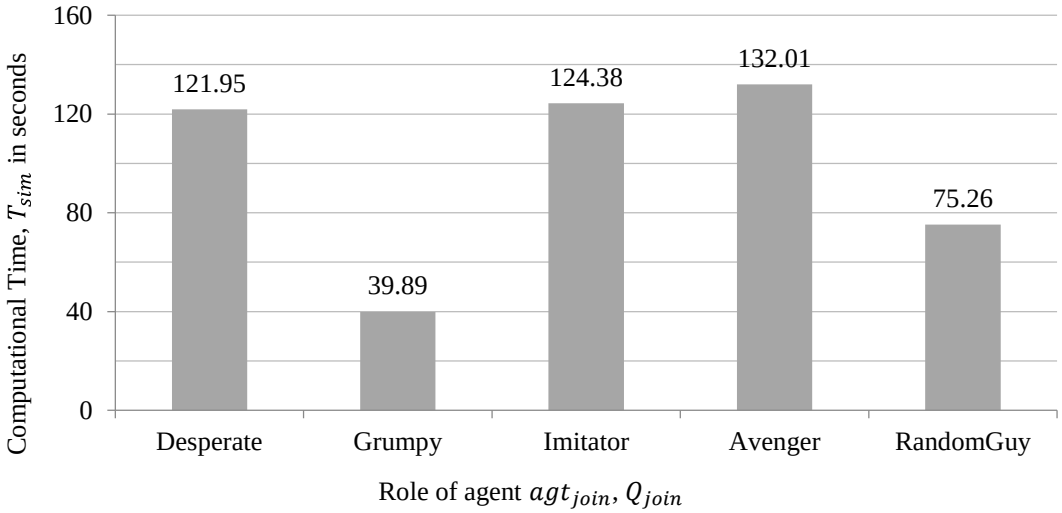


Figure 6.9 Computational time T_{sim} corresponding to agent agt_{join}'s behaviour

From the observation of the Experiment 6.2, we conclude that hypothesis H6.2 is accepted. The agent agt_{join}'s "Desperate" behaviour performed averagely in the NIPD games and it is not efficient in earning scores against coalition members Agt_x with the "Grumpy" behaviour. A flexible cooperative behaviours such as Q_{join}={ "imitator", "avenger"} perform better in countering defective strategy. Thus, we can determine that agent agt_{join} with a flexible cooperative strategy behaviours has a higher chance to join the targeted coalition.

6.4.3 Discussions

In this subsection, other possible issues of agent agt_{join} joining the targeted coalition using JCM are discussed. The JCM encountered some potential limitations such as agent agt_{join} tries to join a coalition but coalition member does not provide anything in return. This scenario shows one-sided contribution is possible during the joining proposal by the agent agt_{join}. The agent agt_{join} is required to decide whether it should join the targeted coalition to achieve its goal g_{join} or join the other compatible coalitions. A negotiation protocol can be implemented to resolve this issue. If an agent agt_{join} wants to achieve its goal without anything in return, it can choose to do so.

Another issue we did not address in this chapter is the potential counter offer by other coalitions. The agent agt_{join} has interest in joining the coalition that can earn more profit. However, the "betrayal" behaviour of agent agt_{join} may affect its reputation in MAS society in the long term. The other concern of the JCM includes the security issue regarding the access of agent agt_{reg}'s databases. If databases have been compromised, it can pose a threat to every agent. To resolve a such problem, counter measures can be implemented such as strict monitoring on databases and setting up an incident response plan for mitigating the lost.

Other potential issue includes multiple offers from different agent agt_{join}. Since the proposed JCM only allows one agent to one coalition interaction by labelling the coalition as "busy". It is possible that multiple offers could be a competitive scenario and agents will fight for the vacancy in the coalition.

6.5 SUMMARY

Based on experimental results, we have shown that joining agent agt_{join} with smart cooperative approach has a better opportunity in joining the targeted coalition. Even though the JCR has been approved by the majority of coalition members $Agt_x \in z$, it does not imply agent agt_{join} will definitely join the coalition. The effectiveness of joining coalition once again heavily relies on the experience between agent agt_{join} and coalition members through the two-phase of voting and trial joining session.

Chapter 7 Democracy based Microscopic Coalition

7.1 INTRODUCTION

Although CF provides agents a domain to increase their utility, the *idle agents* are not able to gain benefit from it. Majority of researchers agree that the increasing the individual agent welfare will indirectly improve the overall society welfare. However, they only emphasize on the individual perspective which involved singular agent in group activities such as coalition. Idle agents have been not able to join the coalition based on two reasons: (1) they did not receive invitation from the coalition organizer, (2) they failed the bid during CFP or (3) they are unable to create a coalition by themselves. This chapter deals with increasing utility of the idle agents by joining a democracy based microscopic coalition through JCM. The idle agents are able propose join coalition request (JCR) to the targeted microscopic coalition. In this chapter, the democracy based microscopic coalition contains a leader agt_{lead} that manages organization affairs. The JCM includes a two-phase evaluation process which involves coalition leaders' evaluation and coalition member's voting.

The voting algorithms serve as an essential part in the decision making for agents in the society. In MAS, weight voting games based voting system is common because agents do not have the same contribution (Zuckerman, Faliszewski, Bachrach, & Elkind, 2012). According to Pitt, Kamara, Sergot, and Artikis (2006), there are several aspects in a robust voting systems such as achieving optimal outcome, systems' flexibility and panel's fairness. These aspects have been the main goals of

the MAS voting model from different approaches (Hemaspaandra & Hemaspaandra, 2007; Parhami, 1994; Pitt et al., 2006; Procaccia, Rosenschein, & Zohar, 2007). A formal voting algorithms use the *Robert's Rules of Order* (*Newly Revised*) (*RONR*) (Robert III et al., 2011) as the voting protocols. It consists of a few steps as shown below using the CF as an example:

- The chair opens the meeting.
- An agent proposes a motion.
- Another agent seconds the motion.
- The agents debate the motion.
- The chair calls agents in favour to cast their vote.
- The chair calls agents who are against to cast their vote.
- The motion is carried or not depending on the standing rules.

Among existing voting models, the *weighted voting game* (WVG) is often used in addressing the decision making model of agents of different inequality. It originated from game theory and is used when voters have different equality in their votes. The main philosophy behind WVG is treating all voters as equal is not always appropriate. According to Itai (1995), a simple WVG consists of a pair of tuple $WVG = (N, \Theta)$ where the Θ represents the agent's weighted vote while N denotes the set of agents that are enfranchised. The quota of the weighted voting systems has been studied by Zuckerman et al. (2012) which is also known as threshold voting. Through manipulation of quota, an agent's power indices can be altered by the central authority. Also, they have noted choosing an optimal quota is harder than NP if it falls into *probabilistic polynomial* (PP) complexity. To measure an agent's power

indices, several measurement models have been proposed which are Shapley-Shubik index (Lloyd S Shapley & Shubik, 1954), Banzhaf index (Banzhaf III, 1964) and Deegan Packel index (John & W, 1978).

The weighted voting model has been implemented into the JCM to help a macroscopic coalition evaluate the agent agt_{join}'s proposal. Agents in coalition have different contributions, thus contribute different weighted votes. The preferential weighted voting is used in the JCM and each coalition members agt_x will allocate vote according to its preference. The vote session halts when the accumulated vote reaches the quota. This process of voting is known as *weighted voting session* (WVS) and the formal term is known as a motion. This chapter is organized as the following: Next section contains the mathematical notations of JCM for a microscopic coalition followed by the algorithms and computational complexity of the algorithms. Next, the experimental methodology and results will be presented. Lastly, the last section summarise this chapter.

7.2.1 MATHEMATICAL NOTATIONS

First, we discuss the weight vote for representing an individual agent's preference during decision making. Next, different categorization of weighted votes in WVS are explained followed by agents' votes allocated by their preference. Subsequently, the quota of WVS is described.

Definition 7.1: Weighted vote Θ denotes an agent decision based on its preference.

The weighted vote of a coalition member consists of a pair of data which are agree Θ_{yes} and disagree vote Θ_{no}. It is based on the preferential vote and agent agt_x will

allocate its preference based on the grading of criteria C_e. The grading of criteria for generating an agree vote is calculated using the following equation:

$$\Theta_{yes} = (con_x) \times (\frac{\sum_{e=1}^{y} C_e}{y}) \tag{7.1}$$

where con_x represents agent agt_x's contribution and C_e represents the criteria of evaluating agent agt_{join}. The con_x is calculated using coalition member agt_x's contribution in a coalition by using the percentage ratio. There are y number of criteria for agent agt_x to evaluate and each criteria is bound by the value of 0.0 $\leq C_e \leq 1.0$. The $\sum_{e=1}^{y} C_e$ shows the evaluation scores and will be converted into ratio of $0.0 \leq \frac{\sum_{e=1}^{y} C_e}{y} \leq 1.0$ to decide on agree and disagree vote. It is based on the preferential voting where agents can allocate its decision into a pair of votes. The following definitions are the agree and disagree vote:

Definition 7.2: Agree vote Θ_{yes} represents coalition member agt_x's vote of supporting agent agt_{join}'s JCR.

The agree vote Θ_{yes} represents the vote of coalition member agt_x allocated for supporting agent agt_{join}'s JCR in a WVS. It is calculated using the Equation 7.1.

Definition 7.3: Disagree vote Θ_{no} represents coalition member agt_x's vote allocated for not supporting agent agt_{join}'s JCR

The disagree vote Θ_{no} represents vote of coalition member agt_x that are allocated for disagreement on agent agt_{join}'s JCR in a WVS. It is calculated using the following Equation 7.2:

$$\Theta_{no} = con_x - \Theta_{yes} \tag{7.2}$$

Example 7.1: Consider the following criteria: goals compatibility C_{join}, budget C_{budget} and trust C_{trust}. If the C_{goal}=0.6, C_{budget}=0.7 and $C_{trust} = 0.8$, the total criteria for consideration is y=3. Hence, to calculate the agree vote Θ_{yes} given agent agt_x's contribution is con_x=1.0, the Equation 7.2 has been applied. If the $\sum_{e}^{y} C_e = 3.0$, this denotes coalition member agt_x fully support agent agt_{join}'s JCR. The agent agt_x's agree vote is $\Theta_{yes} = (1.0) \times \frac{2.1}{3.0}$ where Θ_{yes}=0.7 and the Θ_{no}=0.3.

Definition 7.4: Quota Ω is the minimum vote required to conclude decision making. The quota refers to the minimum votes that are required for a WVS to pass or fail. According to Parhami (1994), there are three standard quotas voting subschemes which are shown as following:

Unanimity voting: $(\Omega = \sum_{x}^{n} \Theta_x) = \sum_{x}^{n} \Theta_{yes}$

Byzantine voting: $(\Omega = (\frac{2}{3} \times \sum_{x}^{n} \Theta_x)) < \sum_{x}^{n} \Theta_{yes}$

Majority voting: $(\Omega = (\frac{1}{2} \times \sum_{x}^{n} \Theta_x)) < \sum_{x}^{n} \Theta_{yes}$

The quota proposed in the WVS is the majority voting where a decision's acceptance is determined by half of agents' number in the coalition. There are two terminating conditions of the WVS which are presented in the Table 7-1:

Table 7-1 WVS Terminating Conditions

Terminating Conditions	***Explanation***
$\sum_{x=1}^{n}\Theta_{yes} \geq (\frac{1}{2}\times\sum_{x=1}^{n}\Theta_{x})$	The agt_{join}'s JCR has passed the WVS.
$\sum_{x=1}^{n}\Theta_{no} \geq (\frac{1}{2}\times\sum_{x=1}^{n}\Theta_{x})$	The agt_{join}'s JCR has failed the WVS.

7.3 APPROACH

7.3.1 Weighted Voting Mechanism

The ideology behind the JCM is to help idle agents increase their utility by joining an existing democracy based microscopic coalition. In this chapter, we will utilize the external description and agents' private information to elaborate the agent agt_{join} joining process. Figure 7.1 shows the enhanced JCM architecture for microscopic coalition.

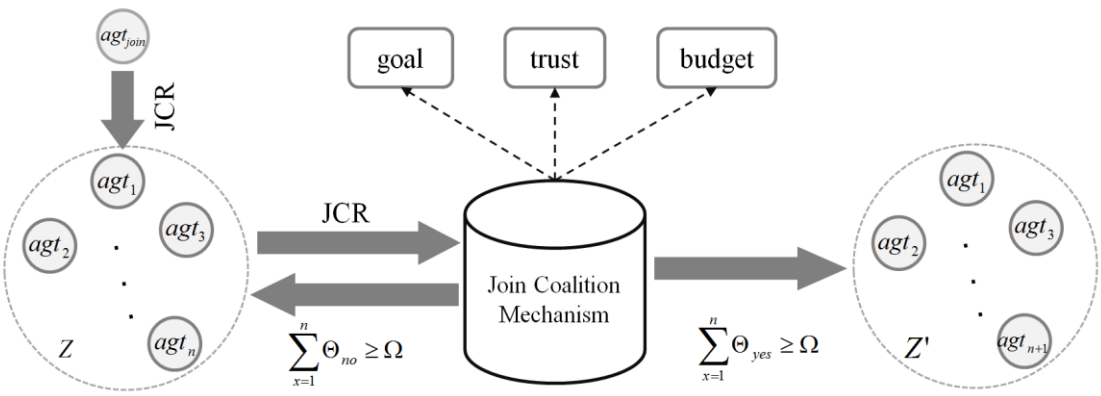


Figure 7.1 The agent architecture with the implementation of the JCM

The agent agt_{join} will initiate the joining coalition event when it has goals that cannot be achieved through the traditional CF mechanism. There are three important criteria for consideration in the proposed models which are the number of the common goals, the trust metrics and the budget evaluation.

First, the coalition Z may consists up to n number of agents and the common goals between agent agt_{join} and targeted coalition z is $\exists(G_{join}) \cap G_z$. The following equation is the coalition members of the targeted coalition without agent agt_{join}:

$$Z = \{agt_1, agt_2, ..., agt_n\}$$

After agent agt_{join} has initial intention of joining a coalition, it will send its JCR to the targeted coalition through agent agt_{reg}. The coalition representative agt_{lead} will suspend current tasks and evaluate the agent agt_{join}'s JCR. If agent agt_{lead} approves agent agt_{join}'s JCR, agent agt_{reg} will organize a WVS for coalition member agt_x to decide on the joining of agent agt_{join}. Otherwise, agent agt_{lead} will reject agent agt_{join}'s proposal and resume suspended tasks. The following equation is used for evaluating the WVS.

$$\sum_{x=1}^{n} \Theta_{yes} \equiv \begin{cases} 0.0 \\ 0.1 \\ . \\ . \\ . \\ 1.0 \end{cases} \text{ if } \sum_{x=1}^{n} \Theta_{yes} \geq \Omega \text{ then } Z = Z'. \text{ Otherwise } Z.$$

The agent agt_{reg} becomes the panel of the WVS and monitors the voting process. Once the disagree or agree votes reach quota, agent agt_{reg} will announce the results to coalition members Agt_x and agent agt_{join}. Upon the majority acceptance of agent agt_{join}'s JCR, agent agt_{join} becomes a part of the coalition and the number of agents increases by one. The coalition Z will undergo a reformation to form a new coalition Z' as shown as below:

$$Z' = \{agt_1, agt_2, ..., agt_n, agt_{n+1}\}$$

If agent agt_{join} received majority disagree vote for its JCR, it will join an alternate coalition that has the next highest compatibility with its goal G_{join}. Agent agt_{join}'s targeted coalition will remain the same which is Z. The agent agt_{join} will revise its knowledge so that it will not seek to join the coalition Z for a certain period.

The JCM algorithms are divided into four parts based on agent's role. Each role holds a different approach in handling with agent agt_{join}'s JCR. We have proposed the different algorithms based on different agents' roles followed by the computational analysis of each algorithm.

7.3.2 Joining Agent's Perspective

The agent agt_{join} is the main character in the JCM and it initializes the event of the joining coalition. Figure 7.2 shows the algorithm for the agent agt_{join} in the JCM. As shown on Chaper 2.3.1, agent agt_{join} has three conditions to be fulfilled before proposing the JCR to the targeted coalition.

Algorithm 7.1 takes place when the agent agt_{join} searches for a compatible coalition through agent agt_{reg}. The compatible coalition is the organization that shares the majority of the common goals $\exists g_{join} \cap G_z$. The agent agt_{join} will receive the LOC from the agent agt_{reg} that is sorted according to the compatibilities percentage of the goals. Based on the LOC, agent agt_{join} will propose its JCR to the targeted coalition z through the agent agt_{reg}. After that, agent agt_{join} will wait for the response from the agent agt_{reg} for the first evaluation result. If it does not receive the response within the interval, it will revise its knowledge and proceed to the next highest compatible coalition. Subsequently, it will wait for the second evaluation of the targeted coalition which is the WVS among the members. If the WVS' result shows the majority coalition members against agent agt_{join}'s JCR, it will revise its knowledge and proceed to the next compatible coalition. Otherwise, if the WVS of agent agt_{join}'s JCR is successful, it wait for agent agt_{lead} to perform coalition reformation.

ALGORITHM 7.1. Agent agt_{join} joins the targeted coalition

1. Requests the LOC from agent agt_{join} that are compatible with its goals.
2. Retrieves the LOC from agent agt_{reg}.
3. Searches the highest compatible coalition from the LOC.
4. Proposes the JCR and the targeted coalition to agent agt_{reg}.
5. Waits for agent agt_{reg}'s respond for a certain interval.
6. If agent agt_{join} did not receive agent agt_{reg}'s respond within the given interval or acknowledgement of JCR's failure, it will search for other compatible

(See next page)

coalition and repeat Step 1. Otherwise, the agent agt_{join} will proceed to Step 7.

7. Waits for the WVS's result through agent agt_{reg}.
8. If $\sum_{x=1}^{n} \Theta_{no} \geq \Omega$, it will revise its belief and repeats Step 1. Else, it will wait for agent agt_{reg} announcement and waits for coalition reformation.

Figure 7.2 Algorithm based on agent agt_{join}'s perspective

7.3.3 Coalition Registration Agent's Perspective

The agent agt_{reg} is the middleman that handles the joining process of the agent agt_{join} with the targeted coalition (note: the definition of the CRA agt_{reg} can be found at Chapter 2). Also, it serves as the intermediary for agents to register their coalition. The agent agt_{reg} earns its incentives by helping idle agents to form or join a coalition. Figure 7.3 shows the algorithm of agent agt_{reg} in the joining process.

ALGORITHM 7.2. Agent agt_{reg} handles agent agt_{join}'s JCR

1. Receives the requests from agent agt_{join} with the goal g_{join}.
2. Checks agent agt_{join}'s status and ensure it is not engaged in any recent coalition activities.
3. Searches and generate LOC for the agt_{join}.
4. Receives agent agt_{join}'s JCR and coalition of its choice.
5. Forwards agent agt_{join}'s JCR to the targeted coalition will be labelled as "busy"
6. Waits for agent agt_{join}'s evaluation result. If the evaluation result shows agent agt_{lead} approve agent agt_{join}'s JCR, notify agent agt_{join} and proceed to Step 7. Otherwise, acknowledge agt_{join} about the rejection of JCR.
7. Notifies agents Agt_x to evaluate and organize the WVS.

(See next page)

8. If $\sum_{x=1}^{n}\Theta_{yes} \geq \Omega$ or $\sum_{x=1}^{n}\Theta_{no} \geq \Omega$, terminate WVS. Otherwise, continue WVS until terminating condition fulfilled.
9. Announces the WVS's results to agent agt_{join} and coalition Agt_x.
10. If $\sum_{x=1}^{n}\Theta_{yes} \geq (\sum_{x=1}^{n}\Theta_x \times \Omega)$, agent agt_{reg} will inform agent agt_{join} to join the coalition. Otherwise, it will inform agent agt_{join} to join an alternate coalition.
11. Updates databases if agent agt_{join} joined the targeted coalition.

Figure 7.3 Algorithm based on agent agt_{reg}'s perspective

The agent agt_{reg} received the request from an agent agt_{join} and it will verify agent agt_{join}'s status. It has to ensure that agent agt_{join} is not engage with any activity to prevent potential spamming attack. If agent agt_{join}'s status is clean, the agent agt_{reg} will search the databases and generate the sorted LOC based on the goal g_{join}. The LOC is sorted according to the agent agt_{join}'s goals compatibilities and sent to agent agt_{join} to decide the targeted coalition. The agent agt_{reg} will wait for agent agt_{join}'s decision on the coalition of its choice. It will forward the agent agt_{join}'s JCR to the targeted coalition for further evaluation. The agent agt_{reg} will label the agent agt_{join}'s targeted coalition as "busy". Upon receiving the first evaluation result, the agent agt_{reg} will notify agent agt_{join} about the result. If the first phase evaluation shows the agent agt_{lead} agree on agent agt_{join}'s JCR, it will acknowledge both parties and prepare for the WVS. Otherwise, agent agt_{reg} will inform the agent agt_{join} to revise its belief and choose an alternate coalition to join. The agent agt_{reg} will monitor the WVS to determine the acceptance of the majority coalition members. Once terminating condition has been fulfilled, agent agt_{reg} will stop counting the

vote and inform both parties about the WVS outcome. If WVS about agent agt_{join}'s JCR is success, agent agt_{reg} will update its database regarding the newly formed coalition. Otherwise, it will continue and resume its suspended tasks.

7.3.4 Coalition Agents' Perspective

The coalition consists of the members such as the representative of the organization agt_{lead} and the other members Agt_x. The event of the JCM starts with the coalition receiving a notification from the CRA agt_{reg} about the agt_{join}'s JCR. The coalition will undergo a two-phase evaluation of agent agt_{join}'s JCR to determine its trustworthiness. The agent agt_{lead}'s algorithm is shown in Figure 7.4 while coalition member Agt_x's algorithm is presented in Figure 7.5.

Agent agt_{lead} plays an important role in evaluating the agent agt_{join}'s JCR before proceeding to the WVS. It is the coalition leader's responsibility to ensure the organization will benefit when evaluating agent agt_{join}'s JCR. The algorithm starts when it receives the notification from agent agt_{reg} regarding the agent agt_{join}'s JCR. Subsequently, agent agt_{lead} will forward agent agt_{join}'s information to the coalition members agt_x for pre-evaluation before commencing the WVS. The agent agt_{lead} will evaluate agent agt_{join}'s JCR and reply its decision to the agent agt_{join} given a limited time. If agent agt_{lead} disagrees with agent agt_{join}'s JCR, it will send a reject notification to agent agt_{reg} and resume its operation. Otherwise, the WVS will be organized and agent agt_{lead} will send its pair of votes $\{\Theta_{no}, \Theta_{yes}\}$ to agent agt_{reg}. After that, it will wait for agent agt_{reg}'s announcement on the WVS's result. If

majority coalition agree on agent agt_{join}'s JCR, agent agt_{lead} will reform the coalition by including agent agt_{join}. Otherwise, coalition will resume its suspended tasks and assign new tasks for agent agt_{join}.

ALGORITHM 7.3. Coalition representative agt_{lead}'s perspective of handling JCM.
1. Receives the notification from agent agt_{reg} about agent agt_{join}'s JCR and notify coalition members $Agt_x \in z$. 2. Agent agt_{lead} evaluates agent agt_{join}'s JCR. 3. If agent agt_{lead} has accepted agt_{join}'s JCR, it will proceed to Step 4. Otherwise, agent agt_{lead} will reject agent agt_{join}'s JCR and resume its operation. 4. Requests a WVS with coalition members $Agt_x \in z$ through agent agt_{reg}. 5. Sends its vote to agent agt_{reg}. 6. Waits for agent agt_{reg}'s announcement on the WVS's result. 7. If $\sum_{x=1}^{n} \Theta_{yes} \geq \Omega$, agent agt_{lead} will reform the coalition by including agent agt_{lead} and the coalition will resume its operation. Otherwise, its operation will be resume without coalition reformation.

Figure 7.4 Algorithm based on agent agt_{lead}'s perspective.

The agent agt_x will be the second judge after the initial acceptance of agent agt_{join}'s JCR. The algorithm for agent agt_x's perspective starts by receiving the notification from the coalition representative agt_{lead}. The agents agt_x in the coalition will then evaluate agent agt_{join}'s JCR based on the goal compatibilities, trust and budget. The weighted vote for the WVS is generated in a pair which are $\{\Theta_{no}, \Theta_{yes}\}$. To maximize an agent agt_x's utility, it will resume existing tasks until receiving the call

for WVS notification from agent agt_{reg}. Upon receiving agent agt_{reg}'s notification, it will suspend current tasks and evaluate agent agt_{join}'s JCR. After evaluating, agent agt_x will send its vote to agent agt_{reg} and resume the existing tasks. The coalition members Agt_x will wait for the result of the WVS for the agent agt_{join}. If the WVS's result show agent agt_{join}'s JCR is accepted by the majority coalition members, it will wait for the coalition reformation order from agent agt_{lead}.

ALGORITHM 7.4. Coalition member agt_x's perspective of handling JCM.

1. Receives the notification from agent agt_{lead} about agent agt_{join}'s JCR.
2. Evaluates agent agt_{join}'s JCR then allocate by preference to the Θ_{no} and Θ_{yes}.
3. Resumes existing tasks until received WVS notification from agent agt_{reg}.
4. Suspends current tasks and send Θ_{no} and Θ_{yes} to agent agt_{reg}.
5. Resumes existing tasks until receiving the WVS's result notification from agent agt_{reg}.
6. If WVS passed, waits for the coalition reformation order from agent agt_{lead}. Otherwise, resume the existing operation.

Figure 7.5 Algorithm based on agent agt_x's perspective.

7.3.5 Computational Complexity Analysis

The computational complexity for Algorithm 7.1 is O($\gamma \times \Re$) where the γ represents number of coalitions registered in database. The $\Re$ represents number of coalitions that have common goals with agent agt_{join}, $\sum_{h=1}^{\Re}(g_h \in G_{join})$. The highest computational complexity occurs in the Step 3 in Algorithm 7.1 where it is required to search through the LOC with the $\Re$ number of compatible goals. The other

computational complexity such as communication and waiting interval are not covered.

The computational complexity for Algorithm 7.3 is O($\Re$). The Step 2 has a major impact on the computational complexity since the worst case for evaluation of $\Re$ goals is O($\Re$). The complexity of Algorithm 7.4 is O($\Re$) which is similar with Algorithm 7.3. Despite having a similar computational complexity, the role of agent agt_x and agt_{lead} is clearly diverse from each other. The agent agt_{lead} performs full phase evaluation where agent agt_x only involves during the second evaluation. In Algorithm 7.4, the Step 2 has the most impact in the computational complexity since it involves the searching of compatible goals through the $\Re$ number of coalitions.

The computational complexity for Algorithm 7.2 is O($n + (\Re \times \Im \log(\Im))$) where the $\Im$ represents the number of the coalition registered in the databases. There are four steps where the computational complexity is high such as Step 2, Step 3, Step 7 and Step 11. The Step 2 consists of O($\Im$) complexity where agent agt_{reg} searches the database for agent agt_{join}'s status. The Step 3 has the complexity of the O($\Re \times \Im \log(\Im)$) as it retrieves $\Re$ number of coalition from database and generate LOC using the sorted linked list with a computational complexity of O($\Im \log(\Im)$). The Step 7 is the WVS and contains n numbers of agents to vote for agent agt_{join}'s JCR. The Step 11 indicates agent agt_{join} has successfully joined the coalition and updating of the latest coalition member lists consume O($\Im$) computational steps (For further information on the complexity, please refer to Appendix F).

7.4 EXPERIMENTAL RESULTS

7.4.1 Performance Measurements and Parameters

The experiment is conducted with a workstation that has the following specification: *Dell Optiplex* 390, *Intel Core i*3 and 4*GB RAM*. The following Table 7-2 shows the simulation parameters:

Table 7-2 Simulation parameters

Parameters	***Description***
$CR_{\Theta yes} = \frac{\sum_{y=1}^{z} C_y}{z}$	The evaluation ratio for determining the vote Θ_{yes} for the coalition member agt_x.
n	The number of coalition members agt_x including the agent agt_{lead}.
con_x	The contribution of coalition member in a coalition agt_x.
Ω	The quota for WVS.

The simulations are conducted using the software created with JAVA language and JUNG (O'Madadhain et al., 2003) library. Figure 7.6 shows an example visual result during JCM simulation.

In Figure 7.6, the different size of agent agt_x represents its contribution in coalition. Every communication between agent agt_{join} and coalition z is done through agent agt_{reg} as shown in previous algorithms. When collected vote has reached its quota, the WVS result will be announced to agent agt_{join} and coalition.

There are two performance measurements for the JCM's simulation which are voting

percentage ϖ_{Θ} and the simulation time T_{sim}. The possible voting result for JCM simulation are incomplete agree vote (IAV), incomplete disagree vote (IDV) and completed vote (CAV). The IAV shows the WVS terminated with a major acceptance of the agent agt_{join}'s JCR where the IDV shows the opposite. The CAV denotes the coalition undergoes the complete WVS with the result of the acceptance. The simulation time denotes the total time for two-phase evaluation which includes the communication between agents in the coalition. It is measured in nanoseconds and then scaled to second for a better readability. The main purpose of this simulation is to study the effect of these parameters: n, $CR_{\Theta yes}$ and Ω on the JCM efficiency. All the simulations are conducted with an iteration of 10,000 executions.

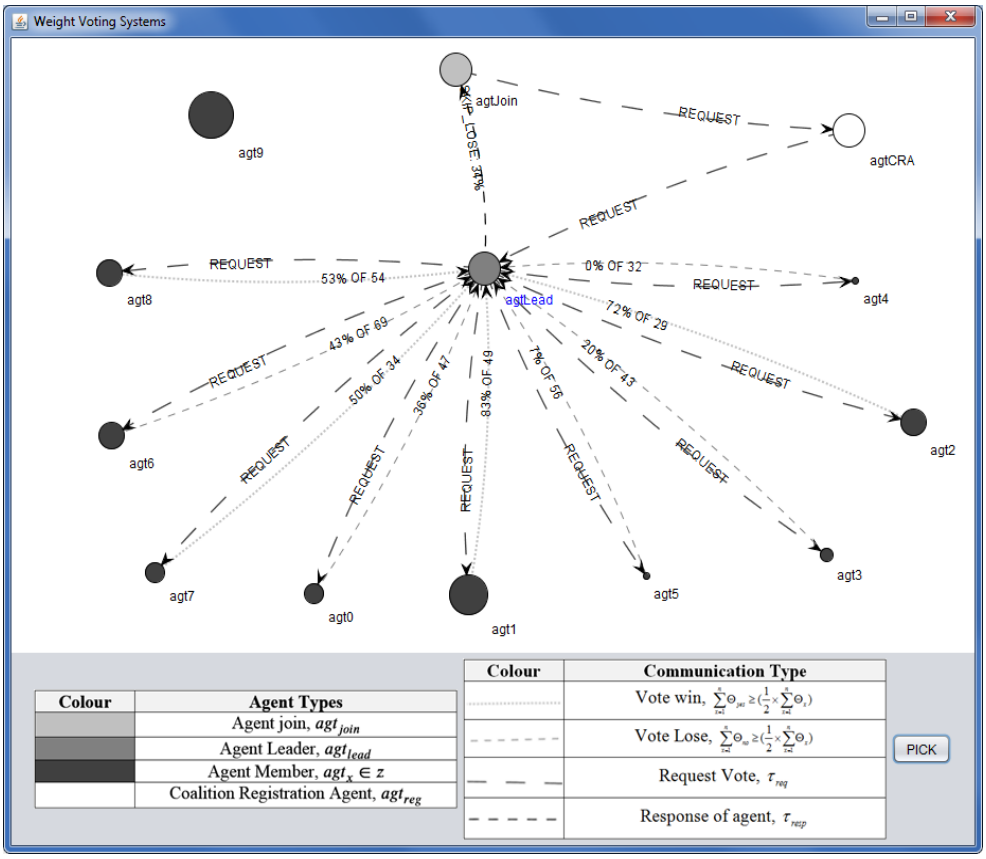


Figure 7.6 The visual example of the simulation using JCM

7.4.2 Experiment 7.1: Number of Agents

In the Experiment 7.1, the impact of number of agents on to JCM efficiency is studied. Before proceeding to the simulation, the following are hypothesis H7.1 which has been derived for conducting the experiments:

*Hypothesis H*7.1: The number of agents n is believed to have a significant impact on the JCM's efficiency and the coalition tends to reject agent agt_{join}'s JCR. The complexity of the WVS increases and it is projected to have a higher competitive WVS. The number of votes $\sum_{x=1}^{n}\Theta_x$ that is required to pass quota Ω is also significantly increased. As suggested by Metcalfe's Law (Shapiro & Varian, 1999), the communication between agents is proportional to the agents' number are n^2. The number of agents n is projected to increase the communication delay between agents.

In order to prove the validity of hypothesis H7.1, the simulation using the parameters as following has been conducted:

$$[CR_{\Theta yes} = 0.5, n = \{50,100,...,500\}, con_x = 60 \pm 30, \Omega = 0.5]$$

The Figure 7.7 shows the WVS result of voting percentage, ϖ_{Θ} corresponding to the number of agents, n. Based on the Figure 7.7, the percentage of the CAV, ϖ_{CAV} has significantly increased from 5.04% to 83.19% when the number of agents n increase. The percentage of IAV ϖ_{IAV} and IDV ϖ_{IDV} have decreased from 43.43% to 8.26% and 51.53% to 8.55% respectively. The main cause behind the decrement of

ϖ_{IAV} and ϖ_{IDV} is the minimum vote to pass WVS's quota has increased when the agents' number is approaching n=500. The agent agt_{reg} is required to collect the votes from every coalition member and this increases the CAV percentage. The increment of ϖ_{CAV} shows the competition for the WVS has increased as more agents will argue for agent agt_{join}'s JCR. Thus, it is observed that the ϖ_{IAV} and ϖ_{IDV} decreases in linear rate while the ϖ_{CAV} has a linear growth rate over the agents' number. The agents' number has increased the ϖ_{CAV} significantly as suggested by the hypothesis H7.1.

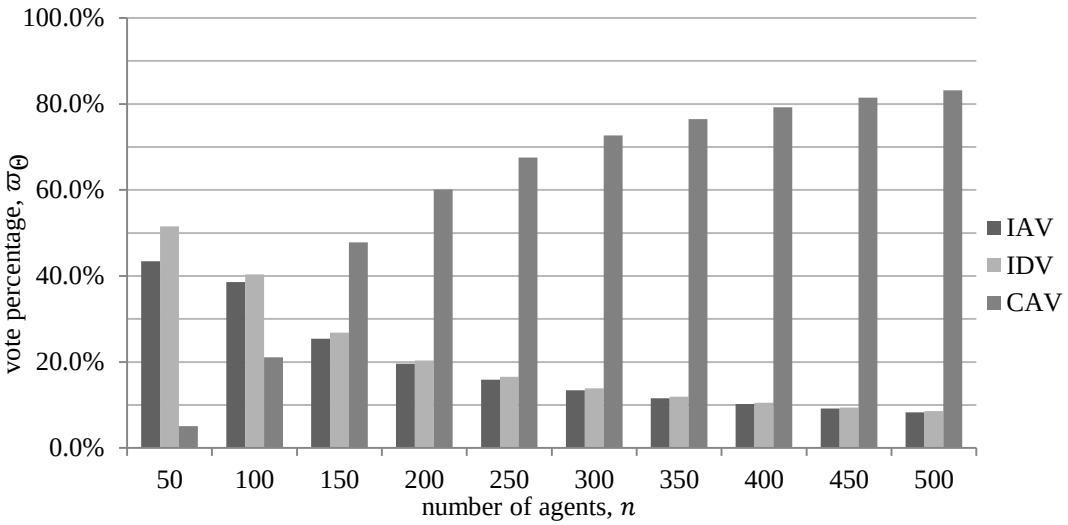


Figure 7.7 The impact of agent's number to voting percentage

The simulation time for the Experiment 7.1 has been recorded to measure the impact of the agents' number over simulation time. The Figure 7.8 shows the simulation time corresponding to the agents' number. It shows the increment of the simulation time from 14.796s to 167.521s. The growth rate of simulation time T_{sim} is linear and it is not exponentially increasing as predicted in Metcalfe's law. The main cause of such phenomenon is the communication only involves agents $agt_x \in z$ through agent

agt_{reg} and it does not require multiple communication for confirmation. Hence, it is not proportional to n^2 and it is increase in linear rate.

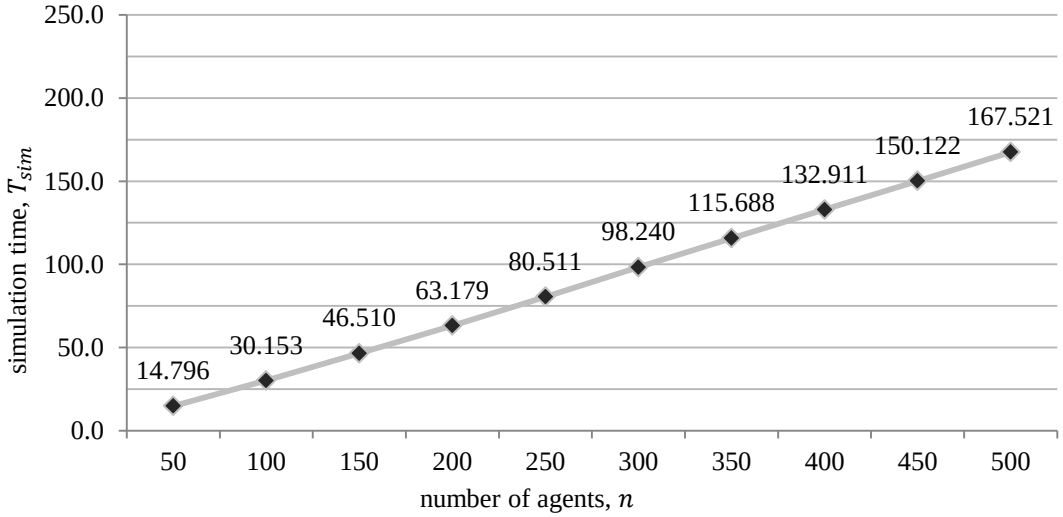


Figure 7.8 Agent's number corresponding to the simulation time

7.4.3 Experiment 7.2: The Evaluation Ratio of Agree Vote

In the Experiment 7.2, the impact of the evaluation ratio of the agree vote on vote percentage and the simulation time are studied. The following hypothesis is derived before the experiment was conducted:

*Hypothesis H*7.2: The high agree vote ratio $CR_{\Theta yes}$ of agent Agt_x is projected to have a higher acceptance rate for the agent agt_{join}'s JCR. We believe that the high evaluation ratio of agree vote $CR_{\Theta yes}$ in the coalition will increase the chances of agent agt_{join} joining an existing coalition. Otherwise, a lower evaluation $CR_{\Theta yes}$ will drive the agents Agt_x to vote for rejecting the agent agt_{join}'s JCR.

The Experiment 7.2 have the simulation parameters as following:

$$[CR_{\Theta yes} = \{0.0, 0.1, ..., 1.0\}, con_x = 50 \pm 25, n = 250, \Omega = 0.5]$$

The Figure 7.9 shows the simulation result based of the agree vote's evaluation ratio corresponding to the vote percentage. When the IAV's vote percentage ϖ_{IAV} is increased from 0.00% until 44.36%, the IDV's vote percentage ϖ_{IDV} has decreased from 71.42% to 45.51%. The ϖ_{CAV} has decreased from 28.58% to 6.25% and slightly increased when it is approaching $CR_{\Theta yes}$=1.0. The main reason behind this phenomenon is the tendency of coalition members Agt_x to support agent agt_{join}'s JCR is increasing. However, the ϖ_{IDV} has significantly decreased over $CR_{\Theta yes}$'s increment because coalition members that agree and vote for the agent agt_{join}'s JCR is increasing. The competition between agree and disagree vote has caused the ϖ_{CAV} decrease over increment of $CR_{\Theta yes}$.

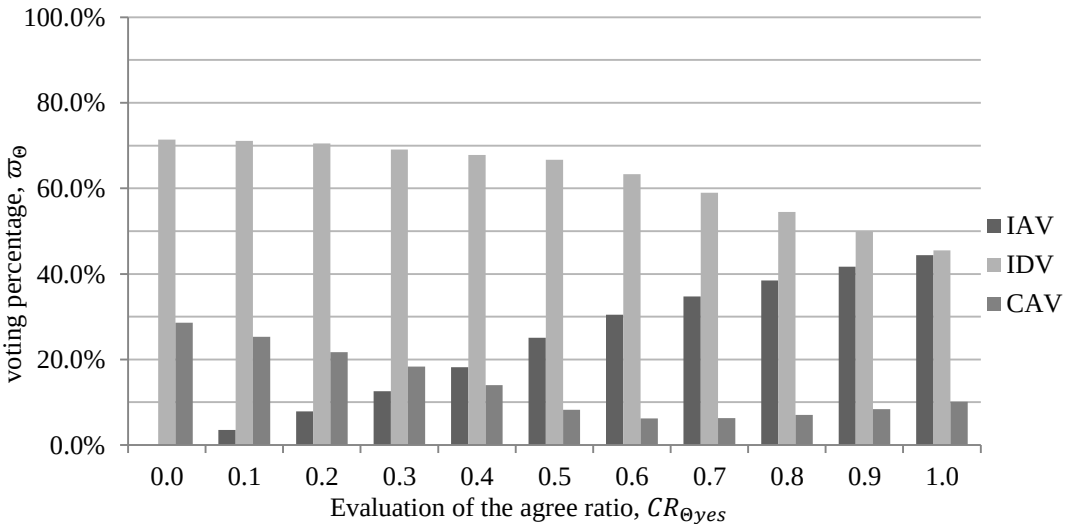


Figure 7.9 The voting percentage corresponding to evaluation on agree ratio

The simulation time for Experiment 7.2 shows a linear growth rate over coalition

member agt_x's evaluation of agree ratio. The simulation time has increased from 15.374s to 170.575s as shown in Figure 7.10. The main reason behind linear growth rate is the coalition member agt_x that supports agent agt_{join}'s JCR is increasing. Hence, the communication during WVS has increased as more coalition members agree on agent agt_{join}'s JCR.

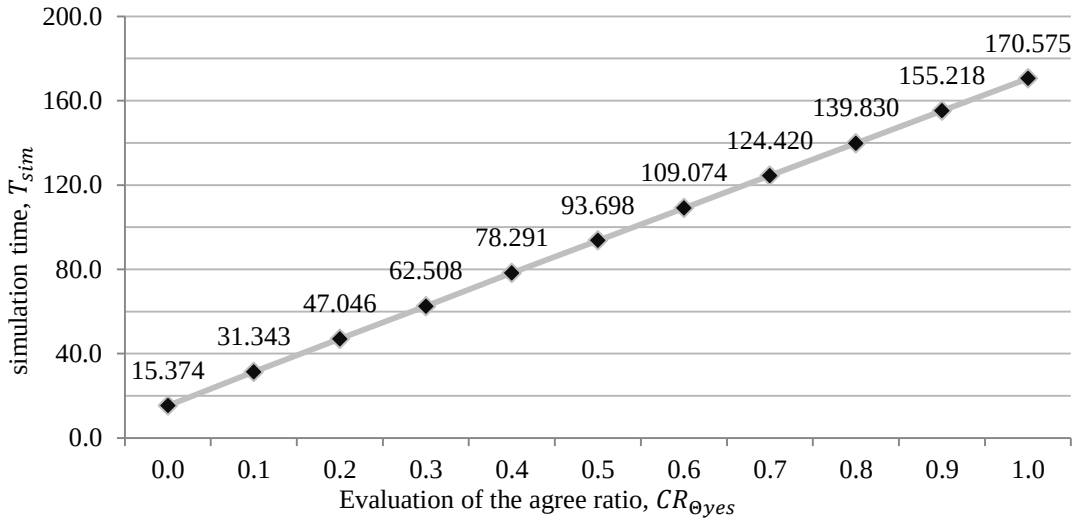


Figure 7.10 The simulation time corresponding to evaluation on the agree ratio

7.4.4 Experiment 7.3: Quota of Weight Voting Session

In the Experiment 7.3, the impact of WVS's quota on the vote percentage and simulation time are studied. The following Hypothesis H7.3 is denoted for Experiment 7.3.

*Hypothesis H*7.3: A higher quota Ω leads to a larger amount of coalition members Agt_x's votes $\sum_{x=1}^{n}\Theta_x$ required to decide the WVS result. We believed that the tendency to have a full session of WVS ϖ_{CAV} will more frequently occur (ϖ_{CAV} will increase) when a higher quota Ω is given. Through the manipulation of quota, WVS result can

be controlled via the minimum vote required.

The simulation for the Experiment 7.3 is conducted with the following parameters:

$$[CR_{\Theta yes} = 0.5, con_x = 60 \pm 30, n = 250, \Omega = \{0.1, 0.2, ..., 0.9\}]$$

Based on the Figure 7.11, the ϖ_{CAV} has decreased significantly from 81.11% to 3.24%. The ϖ_{IAV} and ϖ_{IDV} has increased from 9.30% to 47.87% and 9.59% to 48.89% respectively. Based on the data, the hypothesis H7.3 is rejected where the prediction trend of full WVS is inversed. The ϖ_{IAV} is reduced because the vote required to reach quota has been reduced. The competition between agents have increased as well through the increment of ϖ_{IAV} and ϖ_{IDV}. Both percentages have increased in a linear growth rate corresponding to the quota because total votes required to pass a WVS is increased at the same time.

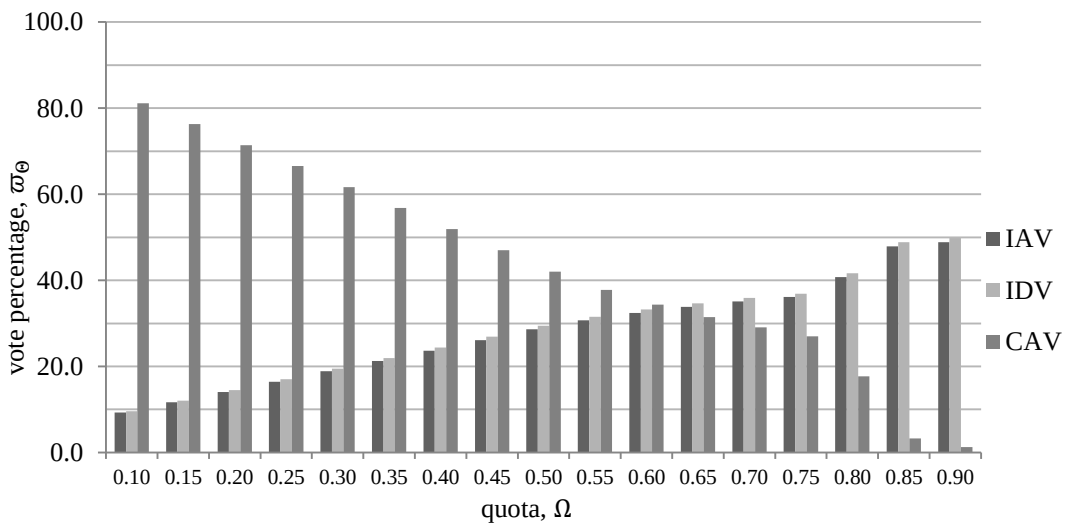


Figure 7.11 The voting percentage corresponding to WVS' quota

Based on the Figure 7.12, the Experiment 7.3's simulation time shows a slow

increasing growth rate when the quota increase at 0.10 until 0.30 and 0.40 until 0.90. When the quota has increased from Ω=0.30 to Ω=0.40, there is a minor increment on simulation time from 10.9s to 29.9s. When quota is at 0.35, the CAV is decreasing and we believed that competition of voting between agents is intensified. Hence, the simulation time has increased significantly when quota is approaching 0.5. On the other hand, the competition is increasing slowly and simulation time is not heavily affected.

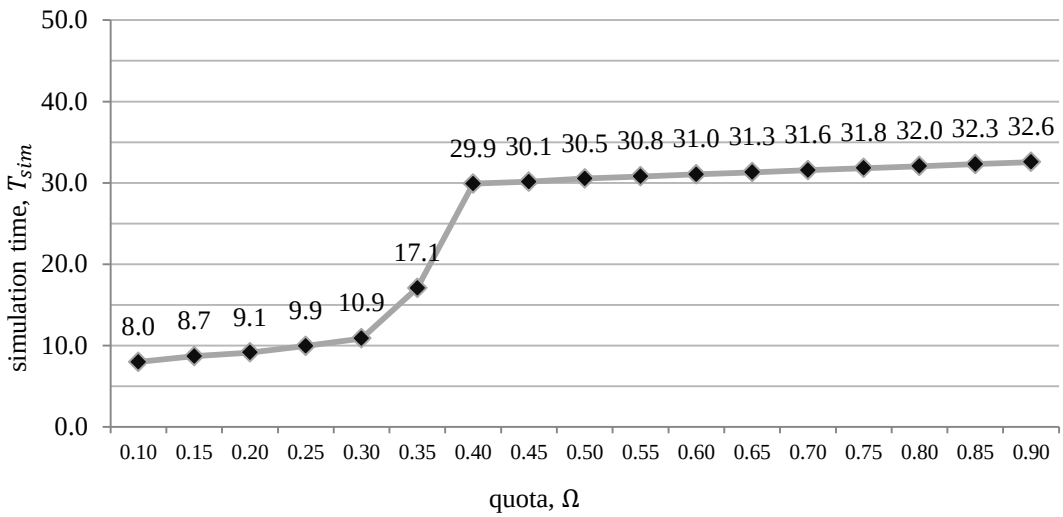


Figure 7.12 The simulation time corresponding to WVS' quota

7.5 SUMMARY

The hypothesis H7.1 is accepted through the observation of simulation result based on Figure 8. The agents' number will significantly increase complexity of WVS to reach an agreement as more votes are required to reach pre-defined quota. The simulation time for Experiment 7.1 shows a growth rate of linear and proofed the communication between agents in WVS is not in exponential. The validity of hypothesis H7.2 is observed through the vote percentage since coalition members Agt_x tends to accept agent agt_{join}'s JCR. Increasing $CR_{\Theta yes}$ has significantly

increased the ϖ_{IAV} where agents Agt_x tends to agree on the agent agt_{join}'s JCR. The simulation time for Experiment 7.2 is increasing in linear rate which is not a huge impact on the time complexity. On the other hand, hypothesis H7.3 is concluded as invalid based on the observation of Experiment 7.3's result. The competition between agents Agt_x has intensified when the WVS' quota has increased. The chances of agent agt_{join} joining the targeted coalition is getting lower due to the increased objection from coalition member agt_x. Increasing the quota will indirectly increase the competition as more votes are required to determine the result of the WVS.

Chapter 8 Conclusion

The interdependence relationsips between agents can be derived from different perspectives such as collaboration between internal and external agents in a coalition. Based on current system design survey, the relationship between agents are not expressed throughoutly as there is room for improvements. By identifying ideal partner, agents can form simpler and efficient cost for maintaining dependence relationships. However, each agent is assumed to have individual knowledge where they do not possess an overview on the relationships at organizational level. In addition, idle agents are unable to increase their utility since they are not engaged in any activity. In order to solve aforementioned problems, we have proposed a heuristic refinement algorithm for CVC, CSG, transitive dependence based coalition and lastly we propose a JCM based on the macroscopic and microscopic coalition.

In Chapter 3, a comprehensive literature review on agents' dependence relationship and CF are presented. Through the dependence theory, we can visualise the dependence relationships between agents via dependence graph. However, current dependence relationships is incomplete in following aspects such as economy, trust, dynamic and decentralisation aspects. On the other hand, CF allows agents to cooperate under an organizational paradigm that is dynamic and goal-oriented (Horling & Lesser, 2004). However, the computational complexity of organizational formation process of coalition is *NP-hard*.

In Chapter 4 , we have proposed T-DepBM and it has shown agents are able to form T-Dep based coalition through the concept of budget. The main focus of T-DepExp

system emphasizes on forming a feasible coalition and indirectly improve an agent's social welfare. By increasing agent's social welfare, the coalition formed can benefits from earning more profit. On the other hands, agents also manage to obtain the organization level view of their relationship by passing the information transitively. T-DepBM has addressed problem with auto-knowledge principle whereas agents only have local view when forming cooperation with other agent. In T-DepBM, information of each child agent agt_i are passed indirectly to root agent agt_{root}. The root agent agt_{root} will perform analysis and obtain an overall view of the coalition struture. Last but not least, our proposed T-DepBM has a computational complexity of $O((^n P_k \times n)\log_i(n))$ which is *NP* complete.

In Chapter 5, we have proposed KDRVM that has shown validation of dependence relationship and achieved with an efficiency of O(E^v). The total accumulating weight and value of coalition are increasing linearly with the number of agents presence in the dependence relationships. On the contrast, the number of interactions between agent agt_{root} and coalition member Agt_x increase in exponential against number of agents in a coalition. Hence, a smaller coalition structure offers a better overall efficiency in communication compare to a larger organization. In the Experiment 5.2, the dep_{or} dependence relationship with a lower weight and value possess the highest communication rate during dependence relationship validation. Despite dep_{or} relationship has a higher communication rate than dep_{1to1} and dep_{and}, but the agent agt_{root} with dep_{or} relationships have options to choose a lower coalition cost.

In Chapter 6, we have proposed JCM that helps idle agents to join an existing macroscopic coalition. It features a double verification modules which includes voting section among coalition for agent agt_{join}'s JCR and conducts NIPD game sessions. Through the simulation, we can observe the ratio of "Grumpy" agents causes winning rate of NIPD games to decrease significantly. However, the communication rate and simulation time has decreased as agent agt_{join} chooses not to join the targeted coalition by terminating JCR proposal. In the second experiment, agents with $Q_{join} =$"imitator" and "avenger" behaviour have a higher score in NIPD games are observed. It also shows agents with a flexible behaviour has a higher chance of winning NIPD games and able to join the existing coalition. Despite JCM allows agent agt_{join} to join an existing coalition within polynomial computational complexity, the coalition reformation process still remains a NP-hardness problem.

The Chapter 7 focuses on extending JCM to the democracy based microscopic coalition. Through implementation of JCM, an agent agt_{join} is able to join the targeted democracy based microscopic coalition using two phase evaluation approach. Various roles of agents have been introduced for assigning tasks during the proposal of the joining agent agt_{join}. The experiments in Chapter 7 shows the agent agt_{join} has a higher chance to join the targeted coalition if the three evaluation criteria in favour of agent agt_{join}. The computational complexity of JCM's algorithms are polynomial but the challenge remain is the computational complexity of coalition reformation process.

In the future works, it would be recommended to investigate combination of trust,

budget and goal (Chapter 4 to Chapter 7) into a real world application. The real world application requires a more dynamic, convergence and faster algorithms in order to get fast and responsive MAS application. We believe JCM is able to perform well in the real world scenario which help coalition one step closer to OMAS. In addition, machine learning and anytime algorithms can further enhance our proposed mechanism such as intelligently reject an external agent's proposal if there is bad reputation or bad experience with it. However, these extensions requires further investigation to prove its efficiency over the existing coalition features.

References

Abdallah, S., & Lesser, V. (2004). Organization-Based Coalition Formation*Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3, held in New York,* New York (pp. 1296-1297). 1018922: IEEE Computer Society. http://dx.doi.org:10.1109/aamas.2004.206

Adel, G., Rajabi, M. H., & Reza, M. (2010, 11-13 May 2010). *A novel algorithm for coalition formation in Multi-Agent Systems using cooperative game theory.* Paper presented at the 2010 18th Iranian Conference on Electrical Engineering, Isfahan, Iran http://dx.doi.org:10.1109/iraniancee.2010.5507017

An, B., Miao, C., & Cheng, D. (2005). A Coalition Formation Framework Based on Transitive Dependence. *IEICE - Trans. Inf. Syst., E88-D*(12), 2672-2680. http://dx.doi.org/:10.1093/ietisy/e88-d.12.2672

An, B., Miao, C., Shen, Z., Miao, Y., & Cheng, D. (2005). Transitive Dependence Based Formation of Virtual Organizations. In Y. Hao, J. Liu, Y. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma & Y.-C. Jiao (Eds.), *Computational Intelligence and Security* (Vol. 3801, pp. 375-380): Springer Berlin Heidelberg. http://dx.doi.org/:10.1007/11596448_54

An, B., Miao, C., Tang, L., Li, S., & Cheng, D. (2005). Toward transitive dependence in MAS*Proceedings of the 6th international conference on Intelligent Data Engineering and Automated Learning, held in Brisbane, Australia,* (pp. 486-493). 2103936: Springer-Verlag. http://dx.doi.org:10.1007/11508069_63

An, B., Shen, Z., Miao, C., & Cheng, D. (2007). Algorithms for Transitive Dependence-Based Coalition Formation. *IEEE Transactions on Industrial Informatics, 3*(3), 234-245. http://dx.doi.org/:10.1109/tii.2007.902255

Ash, R. B. (2012). *Basic probability theory*. United States of America: Dover Publications, Inc.

Axelrod, R. (1980). Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, *24*(1), 3.

Axelrod, R. (2006). The Evolution of Cooperation: Revised Edition. New York: *Basic Books*.

Balke, T., Konig, S., & Eymann, T. (2009). A Survey on Reputation Systems for Artificial Societies. *Technical Report 46 Bayreuth University, 46*, 1-54.

Banzhaf III, J. F. (1964). Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev., 19*, 317.

Blankenburg, B., Dash, R. K., Ramchurn, S. D., Klusch, M., & Jennings, N. R. (2005). Trusted kernel-based coalition formation. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, *held in The Netherlands,* (pp. 989-996). 1082623: ACM. http://dx.doi.org:10.1145/1082473.1082623

Boella, G., Sauro, L., & Torre, L. v. d. (2004). Social Viewpoints on Multiagent Systems. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, *held in New York, New York,* (pp. 1358-1359). 1018941: IEEE Computer Society. http://dx.doi.org:10.1109/aamas.2004.247

Boella, G., Sauro, L., & Torre, L. v. d. (2006). Strengthening Admissible Coalitions. *Proceedings of conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 -- September 1, 2006, Riva del Garda, Italy*, *held in* (pp. 195-199). 1567063: IOS Press.

Boissier, O., & Demazeau, Y. (1994). MAVI: a multi-agent system for visual integration. *Paper presented at the Multisensor Fusion and Integration for Intelligent Systems*, 1994. IEEE International Conference on MFI '94.

Boissier, O., & Demazeau, Y. (1996). ASIC: An architechture for social and individual control and its application to computer vision *Distributed Software Agents and Applications* (pp. 135-149): Springer.

Bomze, I. M., Budinich, M., Pardalos, P. M., & Pelillo, M. (1999). The maximum clique problem *Handbook of combinatorial optimization* (pp. 1-74): Springer.

Bondy, J. A., & Murty, U. S. R. (1976). *Graph theory with applications* (Vol. 290): Macmillan London.

Breban, S., & Vassileva, J. (2002). *A coalition formation mechanism based on inter-agent trust relationships.* Paper presented at the J. C. Proceedings of the first international joint conference on Autonomous agents multiagent systems

Bredin, J., Kotz, D., Rus, D., Maheswaran, R. T., Imer, C., & Basar, T. (2001). A market-based model for resource allocation in agent systems *Coordination of Internet agents* (pp. 426-442). Springer-Verlag London: Springer-Verlag

Bredin, J., Maheswar, R. T., Imer, C., Basar, T., Kotz, D., & Rus, D. (2000). *A game-theoretic formulation of multi-agent resource allocation.* Paper presented at the Proceedings of the fourth international conference on Autonomous agents http://dx.doi.org:10.1145/336595.337525

Cair, P., & Torre, L. v. d. (2009). *Temporal Dependence Networks for the Design of Convival Multiagent Systems (Extended Abstract).* Paper presented at the Proceedings of The th International Conference on Autonomous Agents Multiagent, Systems-Volume 8, Budapest, Hungary

Caire, P., Villata, S., Boella, G., & van der Torre, L. C. (2008). *Conviviality masks in multiagent systems.* Paper presented at the Proceedings of The 7th International Conference on Autonomous Agents multiagent, systems-Volume

7

Carbo, J., Molina, J., & Davila, J. (2003). Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems, 12*(01), 135-155.

Castelfranchi, C. (1998). Modelling social action for AI agents, *Artificial Intelligence 103*(1-2), 157-182. http://dx.doi.org/:10.1016/s0004-3702(98)00056-3

Chalkiadakis, G., & Boutilier, C. (2004). Bayesian Reinforcement Learning for Coalition Formation under Uncertainty. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, *held in New York, New York,* (pp. 1090-1097). 1018877: IEEE Computer Society. http://dx.doi.org:10.1109/aamas.2004.74

Chalkiadakis, G., Markakis, E., & Jennings, N. R. (2012). Coalitional stability in structured environments. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, *held in Valencia, Spain,* (pp. 779-786). 2343808: International Foundation for Autonomous Agents and Multiagent Systems.

Chavez, A., Moukas, A., & Maes, P. (1997). *Challenger: a multi-agent system for distributed resource allocation.* Paper presented at the AGENTS '97 Proceedings of the first international conference on Autonomous agents New York USA. http://dx.doi.org:10.1145/267658.267736

Choi, J.-K. (2007). Trembles may support cooperation in a repeated prisoner's dilemma game. *Journal of Economic Behavior & Organization, 63*(3), 384-393. http://dx.doi.org/:10.1016/j.jebo.2005.07.002

Conte, R., & Sichman, J. S. (2002). Dependence graphs: Dependence within and between groups. *Computational & Mathematical Organization Theory, 8*(2), 87-112.

Cormen, T. H., Stein, C., Rivest, R. L., & Leiserson, C. E. (2001). *Introduction to Algorithms*: McGraw-Hill Higher Education.

Costa, A., & Dimuro, G. (2007). Quantifying degrees of dependence in social dependence relations. *Multi-Agent-Based Simulation VII*, 172-187.

d'Inverno, M., Luck, M., Wooldridge, M., & others, C. I. J. C. O. N. A. I. Ë. (1997). *Cooperation structures*

Dang, V. D., & Jennings, N. R. (2004). Generating Coalition Structures with Finite Bound from the Optimal Guarantees. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, *held in New York, New York,* (pp. 564-571). 1018803: IEEE Computer Society. http://dx.doi.org:10.1109/aamas.2004.132

Dantzig, T., Mazur, J., & Mazur, B. (2007). *Number: The language of science*: Plume.

Davis, M., & Maschler, M. (1963). The Kernel of a Cooperative Game. *Naval Research Logistics Quarterly, 12*, 223-259.

de Lima do Rego Monteiro, J., & Sichman, J. (2006). PartNET++: Simulating Multiple Agent Partnerships Using Dependence Graphs. *Multi-Agent-Based Simulation VI*, 14-23.

Durfee, E. (2006). Distributed problem solving and planning. *Multi-agent systems and applications, 2086/2006*, 118-149. http://dx.doi.org/:10.1007/3-540-47745-4_6

Fan, X., Wang, R., Sun, B., Sun, S., & Yen, J. (2005, April 18-21, 2005). *Multi-agent information dependence.* Paper presented at the Proceedings of the IEEE International conference on Integration of Knowledge Intensive Multi-Agent Systems. http://dx.doi.org:10.1109/KIMAS.2005.1427050

Feldman, M., & Tennenholtz, M. (2010). Structured coalitions in resource selection games. *ACM Transactions on Intelligent Systems and Technology (TIST), 1*(1), 1-21. http://dx.doi.org/:10.1145/1858948.1858952

Gambetta, D. (1990). *Trust: Making and Breaking Cooperative Relations*: Blackwell Publication.

Garvey, A. J., & Lesser, V. R. (1993). Design-to-time real-time scheduling. *IEEE Transactions on Systems, Man and Cybernetics, 23*(6), 1491-1502.

Gaspar, G., & Morgado, L. (2000). *A Social Reasoning Mechanism Based on a New Approach for Coalition Formation* (41740360). Portugal: Retrieved from http://www.di.fc.ul.pt/biblioteca/tech-reports

Georgios, C., & Craig, B. (2012). Sequentially Optimal Repeated Coalition Formation Under Uncertainty. *Autonomous Agents and Multi-Agent Systems, 24*(3), 441-484. http://dx.doi.org/:10.1007/s10458-010-9157-y

Grossi, D., & Turrini., P. (2010). *Dependence theory via game theory.* Paper presented at the Proceedings of the 9th International Conference on Autonomous Agents Multiagent System, Toronto, CanadaRetrieved from http://staff.science.uva.nl/~grossi/pubblicazioni/grossi10dependence.pdf

Guessoum, Z., Ziane, M., & Faci, N. C. (2004). *Monitoring and organizational-level adaptation of multi-agent systems.* Paper presented at the Proceedings of the Third International Joint Conference on Autonomous AgentsMultiagent, Systems-Volume 3

Guido, B., der, T. L. v., & Serena, V. (2008). *Self Adaptive Coalitions in Multiagent Systems* In *Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO'08.* (pp. 461-462). IEEE.

Guido, B., Torre, L. v. d., & Villata, S. (2009). *Four Measures for the Dynamics of Coalitions in Social Networks.* Paper presented at the Proceedings of the 20th ACM conference on Hypertext and hypermedia

Guye-Vuillème, A., & Thalmann, D. (2000). A high-level architecture for believable social agents. *Virtual Reality, 5*(2), 95-106. http://dx.doi.org/:10.1007/BF01424340

Han, K.-H., & Kim, J.-H. (2000). *Genetic quantum algorithm and its application to combinatorial optimization problem.* Paper presented at the Proceedings of the 2000 Congress on Evolutionary Computation, 2000.

Hemaspaandra, E., & Hemaspaandra, L. A. (2007). Dichotomy for voting systems. *Journal of Computer and System Sciences, 73*(1), 73-83. http://dx.doi.org/:http://dx.doi.org/10.1016/j.jcss.2006.09.002

Hochbaum, D. S. (1997). Approximation algorithms for NP-hard problems. *ACM SIGACT News, 28*(2), 40-52.

Horling, B., & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review, 19*(4), 281-316. http://dx.doi.org/:10.1017/s0269888905000317

Hossein, B. M., Taghi, H. M., Nicole, I., & Hamid, M. (2010). The Cooperative Game Theory Foundations of Network Bargaining Games Automata, Languages and Programming. In S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide & P. Spirakis (Eds.), (Vol. 6198, pp. 67-78): Springer Berlin / Heidelberg. http://dx.doi.org/:10.1007/978-3-642-14165-2_7

Huynh, T., Jennings, N., & Shadbolt, N. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems, 13*(2), 119-154. http://dx.doi.org/:10.1007/s10458-005-6825-4

Itai, S. (1995). Equilibria in weighted voting games with sidepayments. *Journal of Theoretical Politics, 7*(3), 283-300.

John, D. J., & W, P. E. (1978). A new index of power for simplen-person games. *International Journal of Game Theory, 7*(2), 113-123.

Jonker, C., & Treur, J. (1999). Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In F. Garijo & M. Boman (Eds.), *Multi-Agent System Engineering* (Vol. 1647, pp. 221-231): Springer Berlin Heidelberg. http://dx.doi.org/:10.1007/3-540-48437-x_18

Juang, C.-F. (2004). A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34*(2), 997-1006.

Jurisic, M., Kermek, D., & Konecki, M. (2012). *A review of iterated prisoner's dilemma strategies.* Paper presented at the MIPRO, 2012 Proceedings of the 35th International Convention

Kahan, J. P., & Rapoport, A. (1984). *Theories of coalition formation*: L. Erlbaum Associates.

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack problems*: Springer

Verlag.

Ketchpel, S. (1994). *Forming coalitions in the face of uncertain rewards.* Paper presented at the AAAI 1994

Klusch, M., & Gerber, A. (2002). Dynamic coalition formation among rational agents. *Intelligent Systems, IEEE, 17*(3), 42-47. http://dx.doi.org/:10.1109/mis.2002.1005630

Komenda, A., Vokrinek, J., & Pechoucek, M. (2011). Plan representation and execution in multi-actor scenarios by means of social commitments. *Web Intelli. and Agent Sys., 9*(2), 123-133. http://dx.doi.org/:10.3233/wia-2011-0210

Kraus, S., Shehory, O., & Taase, G. C. A. (2003). *Coalition Formation with Uncertain Heterogeneous Information*

Lagoudakis, M. G. (1996). *The 0-1 knapsack problem - An introductory survey*. Retrieved from citeseer.ist.psu.edu/151553.html

Lau, B. P. L., Singh, A. K., & Tan, T. P. L. (2012, 6 - 7 November 2012). *T-DepExp: Simulating Transitive Dependence Based Coalition Formation.* Paper presented at the Engineering Goes Green, 7th CUTSE Conference, Miri, Sarawak

Leiserson, C. E., Rivest, R. L., Stein, C., & Cormen, T. H. (2001). *Introduction to algorithms*: The MIT press.

Marsh, S. P. (1994). *formalising trust as a computational concept.* University of Stirling

Martin, D. L., Cheyer, A. J., & Moran, D. B. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence, 13*(1-2), 91-128. http://dx.doi.org/:10.1080/088395199117504

Morton, D., & Michael, M. (1967). The Structure of the Kernel of a Cooperative Game. *SIAM Journal on Applied Mathematics, 15*(3), 569-604.

Muller, G., & Vercouter, L. (2005). Decentralized Monitoring of Agent Communications with a Reputation Model. In R. Falcone, S. Barber, J. Sabater-Mir & M. Singh (Eds.), *Trusting Agents for Trusting Electronic Societies* (Vol. 3577, pp. 144-161): Springer Berlin Heidelberg. http://dx.doi.org/:10.1007/11532095_9

Nardin, L. G., & Sichman, J. S. (2010, 24-25 Oct. 2010). *Simulating the Impact of Trust in Coalition Formation: A Preliminary Analysis.* Paper presented at the Social Simulation (BWSS), 2010 Second Brazilian Workshop on. http://dx.doi.org:10.1109/bwss.2010.12

Nardin, L. G., & Sichman, J. S. (2011). A Study of the Influence of Trust in Coalition Formation. *Proceedings of the 2nd. Conference of the Computational Social Science Society of America (CSSSA'11), held in Santa Fe, New Mexico, USA,*

October 9-12, 2011 (pp. 10). Santa Fe, New Mexico, USA

Nathan, G., & Luck, M. (2003). *Coalition Formation through Motivation and Trust.* Paper presented at the Proceedings of the second international joint conference on Autonomous agents and multiagent systems

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review, 4*(4), 251-277.

O'Madadhain, J., Fisher, D., White, S., & Boey, Y. (2003). The jung (java universal network/graph) framework. *University of California, Irvine, California,*

Osborne, M. J., & Rubinstein, A. (1994). *A course in game theory* (First Edition ed.). London, England: The MIT Press.

Parhami, B. (1994). Voting algorithms. *IEEE Transactions on Reliability, 43*(4), 617-629. http://dx.doi.org/:10.1109/24.370218

Petrie, C., & Bussler, C. (2003). Service agents and virtual enterprises: A survey. *IEEE Internet Computing, 7*(4), 68-78.

Philip, R., Eyal, W., & Myrna, W. (2012). The partnered core of a game with side payments. *Social Choice and Welfare, 39*(2), 521-536. http://dx.doi.org/:10.1007/s00355-012-0651-9

Pistolesi, G., & Modesti, P. (2001). Complexity and State-transitions in social Dependence Networks. *Discrete Dynamics in Nature and Society. Overseas Publishers Association N.V., 6*, 49-56.

Pitt, J., Kamara, L., Sergot, M., & Artikis, A. (2006). Voting in multi-agent systems. *The Computer Journal, 49*(2), 156-170.

Procaccia, A. D., Rosenschein, J. S., & Zohar, A. (2007). *Multi-Winner Elections: Complexity of Manipulation, Control and Winner-Determination.* Paper presented at the IJCAI

Rahwan, T., & Jennings, N. R. (2007). An algorithm for distributing coalitional value calculations among cooperating agents. *Artificial Intelligence, 171*(8–9), 535-567. http://dx.doi.org/:http://dx.doi.org/10.1016/j.artint.2007.03.002

Rahwan, T., Ramchurn, S. D., Jennings, N. R., & Giovannucci, A. (2009). An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research, 34*(2), 521.

Ramchurn, S., Sierra, C., Godó, L., & Jennings, N. R. (2003). . *proceedings of 6th international workshop of deception, fraud and trust in agent societies, held in Siguenza, Spain* (pp. 69-75).

Robert III, H. M., Robert, H. M., Robert, S. C., Evans, W. J., Honemann, D. H., & Balch, T. J. (2011). *Robert's rules of order: Newly revised*: Da Capo Press.

Rodrigues, M., Costa, A., & Bordini, R. (2002). *A System of Exchange Values to*

Support Social Interactions in Artificial Societies

Rodrigues, M. R., & Costa, A. C. d. R. (2003). Using qualitative exchange values to improve the modelling of social interactions *Multi-Agent-Based Simulation III 4th International Workshop, MABS 2003, Melbourne, Australia, July 14, 2003. Revised Papers* (Vol. 2927, pp. 57-72): Springer Berlin Heidelberg. http://dx.doi.org/:10.1007/978-3-540-24613-8_5

Rogers, A., Dash, R. K., Ramchurn, S. D., Vytelingum, P., & Jennings, N. R. (2007). Coordinating team players within a noisy Iterated Prisoner's Dilemma tournament. *Theoretical Computer Science, 377*(1–3), 243-259. http://dx.doi.org/:http://dx.doi.org/10.1016/j.tcs.2007.03.015

Ross, P. E. (2003). 5 Commandments: The rules engineers live by weren't always set in stone. *December 2003 issue, IEEE Spectrum* Retrieved from http://spectrum.ieee.org/semiconductors/materials/5-commandments

Roth, L. S. (1988). *The Shapley Value Essays in honor of Lloyd S. Shapley*: Cambridge University Press.

Rothkopf, M. H., Pekeč, A., & Harstad, R. M. (1998). Computationally manageable combinational auctions. *Management science, 44*(8), 1131-1147.

Sabater, J., & Sierra, C. (2001). REGRET: reputation in gregarious societies. *Proceedings of the fifth international conference on Autonomous agents*, *held in Montreal, Quebec, Canada,* (pp. 194-195). 376110: ACM. http://dx.doi.org:10.1145/375735.376110

Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artif. Intell. Rev., 24*(1), 33-60. http://dx.doi.org/:10.1007/s10462-004-0041-5

Sandhlom, T. W., & Lesser, V. R. T. (1997). Coalitions among computationally bounded agents. *Artificial Intelligence, 94*(1–2), 99-137. http://dx.doi.org/:http://dx.doi.org/10.1016/S0004-3702(97)00030-1

Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions. *Artificial intelligence, 135*(1), 1-54.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence, 111*(1), 209-238.

Schillo, M., BУMrckert, H. J., Fischer, K., & Klusch, M. C. (2001). *Towards a definition of robustness for market-style open multi-agent systems.* Paper presented at the Proceedings of the fifth international conference on Autonomous agents

Sen, S., & Dutta, P. S. (2000, 2000). *Searching for optimal coalition structures.* Paper presented at the Fourth International Conference on MultiAgent Systems, 2000 Proceedings. . http://dx.doi.org:10.1109/ICMAS.2000.858465

Shapiro, C., & Varian, H. R. (1999). *Information Rules: A Strategic Guide To The Network Economy*. Boston, Massachusetts: Harvard Business Press.

Shapley, L. S. (1953). A Value for n-person Games. *Annals of Mathematical Studies v, 2*, 1-13.

Shapley, L. S., & Shubik, M. (1954). A method for evaluating the distribution of power in a committee system. *American Political Science Review, 48*(03), 787-792.

Shehory, O., & Kraus, S. (1996). *A kernel-oriented model for coalition-formation in general environments: Implementation and results.* Paper presented at the AAAI/IAAI, Vol. 1

Shehory, O., & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence, 101*(1–2), 165-200. http://dx.doi.org/:http://dx.doi.org/10.1016/S0004-3702(98)00045-9

Sichman, J., & Demazeau, Y. (1995). Exploiting social reasoning to enhance adaptation in open multi-agent systems. *Advances in Artificial Intelligence*, 253-263.

Sichman, J. S. (1998). Depint: Dependence-based coalition formationin an open multi-agent scenarios. *Journal of Artificial Societies and Social Simulation, 1*(2), Retrieved from http://jasss.soc.surrey.ac.uk/1/2/3.html

Sichman, J. S., & Conte, R. (2002). Multi-agent dependence by dependence graphs*Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, *held in Bologna, Italy,* (pp. 483-490). 544855: ACM. http://dx.doi.org:10.1145/544741.544855

Sichman, J. S., Conte, R., Demazeau, Y., & Castelfranchi, C. (1998). A Social Reasoning Mechanism based on Dependence Networks. In N. H. Michael & P. S. Munindar (Eds.), *Readings in agents* (pp. 416-420): Morgan Kaufmann Publishers Inc.

Sichman, J. S., & Demazeau, Y. (2001). *On Social Reasoning in Multi-Agent Systems*. Revista Ibero-Americana de Inteligência Artificial: M. AEPIA.

Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver., *IEEE Transactions on Computers, 100*(12), 1104-1113.

Sycara, K. P. (1998). Multiagent Systems. *AI Magazine, 19,* 79-92.

Travica, B. (1997). *The design of the virtual organization: A research model.* Paper presented at the Proceedings of the Americas Conference on Information Systems. August

Vokrinek, J., Komenda, A., & Pechoucek, M. (2009). Decommitting in multi-agent execution in non-deterministic environment: experimental approach. *Proceedings of The 8th International Conference on Autonomous Agents and*

Multiagent Systems - Volume 2, held in Budapest, Hungary, (pp. 977-984). 1558148: International Foundation for Autonomous Agents and Multiagent Systems.

Vokrinek, J., Komenda, A., & Pechoucek, M. (2011). Abstract Architecture for Task-oriented Multi-agent Problem Solving. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 41*(1), 31-40.

Wang, B., Diao, M., & Gao, H.-y. (2008). An improved particle swarm optimization algorithm for knapsack problems. *Applied Science and Technology, 3*, 7.

Wojciech, G., & Kwasnicka, H. (2008, 20-22 Oct. 2008). *Coalition formation in multi-agent systems - an evolutionary approach.* Paper presented at the IMCSIT International Multiconference on Information Technology. http://dx.doi.org:10.1109/IMCSIT.2008.4747228

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems - Second Edition* (2 ed.). New York: John Wiley & Sons.

Wooldridge, M., Jennings, N., & Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems, 3*(3), 285-312. http://dx.doi.org/:10.1023/A:1010071910869

Xiong, W. q., Zhou, Y., & Wei, P. (2005). A dynamic ant colony algorithm with disaster. *Journal of Circuits and Systems, 6*, 21.

Xiong, Y., Chen, H.-h., Miao, F.-y., & Wang, X.-f. (2004). A quantum genetic algorithm to solve combinatorial optimization problem. *Acta Electron Sinica, 11*, 1855-1858.

Yang, J., & Luo, Z. (2007). Coalition formation mechanism in multi-agent systems based on genetic algorithms. *Applied Soft Computing, 7*(2), 561-568. http://dx.doi.org/:http://dx.doi.org/10.1016/j.asoc.2006.04.004

Yun Yeh, D. (1986). A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics, 26*(4), 467-474. http://dx.doi.org/:10.1007/BF01935053

Yusen, Z., Jun, W., Chongjun, W., & Junyuan, X. (2012, 7-9 Nov. 2012). *On the Complexity and Algorithms of Coalition Structure Generation in Overlapping Coalition Formation Games.* Paper presented at the Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on. http://dx.doi.org:10.1109/ICTAI.2012.121

Zambonelli, F., Jennings, N. R., & Wooldridge, M. (2003). Developing multiagent systems: The Gaia methodology. *ACM Trans. Softw. Eng. Methodol., 12*(3), 317-370. http://dx.doi.org/:10.1145/958961.958963

Zick, Y., Chalkiadakis, G., & Elkind, E. (2012). Overlapping coalition formation games: charting the tractability frontier. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, held in Valencia, Spain,* (pp. 787-794). 2343809: International Foundation for

Autonomous Agents and Multiagent Systems.

Zilberstein, S., & Russell, S. (1996). Optimal composition of real-time systems. *Artificial Intelligence, 82*(1), 181-213.

Zlotkin, G., & Rosebschein, J. S. (1994). Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains*The Proceeding of Twelve National Conference on Artificial Intelligence*, *held in Seattle, Washington,* August 1-4, 1994 (pp. 432-437). The AAAI Press, Menlo Park, California.

Zuckerman, M., Faliszewski, P., Bachrach, Y., & Elkind, E. (2012). Manipulating the quota in weighted voting games. *Artificial Intelligence-Kidlington, 180*, 1.

Appendices

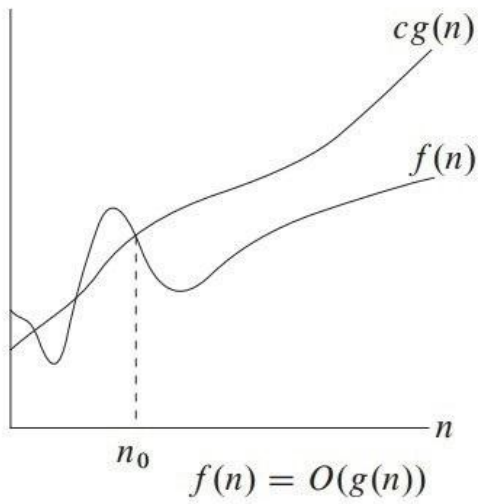
APPENDIX A – COMPUTATIONAL COMPLEXITY - BIG O NOTATION

The computational complexity applied in the paper advocates the worst case scenario and is guarantee the upper bound of the algorithms. The following are the fundamental explanation of the Big O notation that the thesis use.

Big O-notation

For a given function $g(n)$, we denote by $O(g(n))$ the set of functions if there exist positive constant c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

We can say that $g(n)$ is an asymptotic upper bound for $f(n)$.



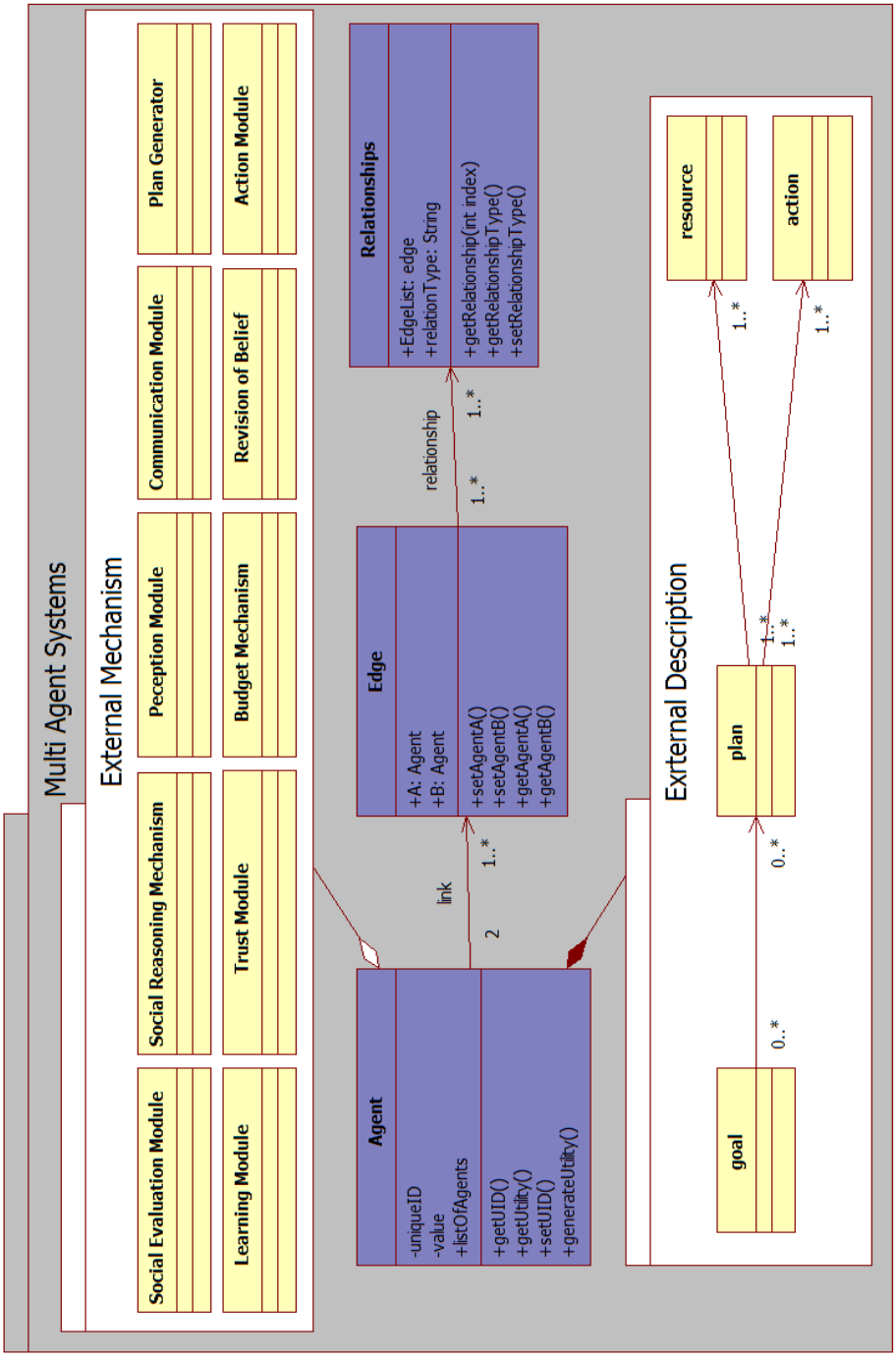
Graphic examples of the Θ, O, and Ω notations.

Taken from Cormen et al., Introduction to Algorithms (Cormen, Stein, Rivest, & Leiserson, 2001).

APPENDIX B – AGENTS' ARCHITECTURE

The agent has the BDI architecture as following:

- External mechanism
- Agent particular information
- Relationship database
- The external description for representing its relationships.



APPENDIX C – CHAPTER 4'S ALGORITHMS AND RESULTS

Computational Complexity Analysis

Algorithm 4.1

ALGORITHM 4.1. The agent agt_{root} checking the validity of the coalition cost.	

Function : GET_FEASIBLE()
Input : *Root_Agent*, *budget_RootAgent*, *Expected_pay*, *counter*
Output : *path*
IF *counter* < *5* **THEN**
CoalitionStatusMessage = GET_COST(*Root_Agent*, *path*, *budget_RootAgent*)
IF *CoalitionStatusMessage* IS *valid* **THEN**
RETURN CoalitionStatusMessage.get_path()
ELSE
counter = *counter* + 1
DISPLAY CoalitionStatusMessage.fail_message()
CHANGE _CRITERIA()
GET_FEASIBLE()
ELSE
RETURN CoalitionStatusMessage.exceed_attempt()
	$O(\phi)$
	$O((^nP_k \times n)\log_i(n))$
	$O(1)$
	$O(1)$
	$O(1)$
	$O(1)$
	$O(1)$
	$O(1)$
	$O(1)$

TOTAL COMPLEXITY	$O(\phi \times ((^nP_k \times n)\log_i(n)))$

Algorithm 4.2

ALGORITHM 4.2. Analyzing the dependence relationship.	
Function : GET_COST()	
Input : *current_agent*, *path*, *coalition_budget*	
Output : *message*	
IF *path* DO NOT CONTAIN *current_agent* **THEN**	O(1)
IF *current_agent* DO NOT HAVE *any children* **THEN**	O(1)
IF *current_agent_cost* WITHIN *coalition_budget* **THEN**	O(1)
ADD *current_agent* INTO path	O(1)
RETURN *success_message*	
ELSE	
RETURN *not_feasible_message*	O(1)
ELSE	
FOREACH *relationship* IN *current_agent* **DO**	
relationshipType = ANALYSE_RELATIONSHIP(*current_agent*)	O(1)
IF *relationshipType* IS "*single-relationship*" **THEN**	O(1)
SINGLE_RELATIONSHIP()	$O({}^{n}P_{k} \times n)$
IF *relationshipType* IS "*and-relationship*" **THEN**	$O(\log_{i} n)$
AND_RELATIONSHIP()	
IF *relationshipType* IS "*or-relationship*" **THEN**	O(1)
OR_RELATIONSHIP()	
ENDFOR	
ELSE	O(1)
RETURN *exist_path_message*	
TOTAL COMPLEXITY	$O(({}^{n}P_{k} \times n) \log_{i} n)$

Algorithm 4.3

ALGORITHM 4.3. Evaluation of the dep_{1to1} relationship	
Function : SINGLE_RELATIONSHIP()	
Input : *current_agent*, *path*, *coalition_budget*, *expected_pay*	
Output : *message*	
IF *current_agent* ADD INTO *path* IS *feasible* **THEN**	O(1)
nextBudget = *budget* – *current_agent.actionCost*	O(1)
ADD *current_agent* INTO *path*	O(1)
temp_message = GET_COST(*child_agent*, *path*, *nextBudget*, *expectedPay*)	O(1)
IF *temp_message* IS *success* **THEN**	
RETURN *success_message*	O(1)
ELSE	
RETURN *not_feasible_message*	O(1)
ELSE	
RETURN *not_feasible_message*	O(1)
TOTAL COMPLEXITY	O(1)

Algorithm 4.4

ALGORITHM 4.4. Evaluation of the dep_{or} relationship	
Function : OR_RELATIONSHIP()	
Input : *current_agent*, *path*, *coalition_budget*, *expected_pay*	
Output : *message*	
IF *current_agent* ADD INTO *path* IS *feasible* **THEN**	O(1)
Candidate = CHOOSEAGENT(*current_agent*, *expected_pay*)	O(1)
nextBudget = *budget* – *current_agent.actionCost*	O(1)
ADD *current_agent* INTO *path*	O(1)
temp_message = GET_COST(*Candidate*, *path*, *nextBudget*, *expectedPay*)	O($\log_i n$)
IF *temp_message* IS *success* **THEN**	O(1)
RETURN *success_message*	
ELSE	O(1)
RETURN *not_feasible_message*	
ELSE	O(1)
RETURN *not_feasible_message*	
TOTAL COMPLEXITY	O($\log_i n$)

Algorithm 4.5

ALGORITHM 4.5. Evaluation of the dep_{and} relationship	
Function : AND_RELATIONSHIP()	
Input : *current_agent*, *path*, *coalition_budget*, *expected_pay*	
Output : *message*	
IF *current_agent* ADD INTO *path* IS *feasible* **THEN**	O(1)
ADD *current_agent* INTO *path*	O(1)
nextBudget = *budget* – *current_agent.actionCost*	
WHILE have_child(*current_agent*) **DO**	O($^{n}P_{k} \times n$)
temp_msg = GET_COST(*get_child*(*current_agent*), *path*, *nextBudget*, *expectedPay*)	O(1)
IF *temp_msg* IS *valid* **THEN**	
num_appropriate = *num_appropriate* + 1	O(1)
END_WHILE	
IF *num_appropriate* EQUALS getNumChild(*current_agent*)**THEN**	O(1)
RETURN *success_message*	
ELSE	O(1)
RETURN *not_feasible_message*	
ELSE	O(1)
RETURN *not_feasible_message*	
TOTAL COMPLEXITY	O($^{n}P_{k} \times n$)

Experimental Results

Experiment 4.1

P_a \ P_{and}	**0.0**	**0.2**	**0.4**	**0.6**	**0.8**	**1.0**
0.20	44.357	44.300	44.440	44.290	44.409	44.350
0.40	44.389	44.476	44.186	44.533	44.29	44.654
0.60	44.238	44.415	44.332	44.379	44.287	44.379
0.80	44.295	44.174	44.305	44.148	44.234	44.421
1.00	44.043	43.956	44.023	43.158	43.587	43.420
1.20	21.826	17.15	12.573	7.960	3.950	0.000
1.40	20.018	16.376	11.638	8.393	3.003	0.000
1.60	17.005	14.798	10.589	6.711	3.044	0.000
1.80	14.344	12.478	9.147	6.197	2.725	0.000
2.00	13.122	10.611	7.620	5.870	2.491	0.000

Experiment 4.2

n	$\sum_{i=1}^{n} U_i$
50	39.84
100	41.13
150	42.59
200	42.70
250	42.94
300	43.38
350	43.27
400	43.17
450	43.49
500	43.46

Experiment 4.3

β_{root}	$\sum_{i=1}^{n} U_i$	$\sum_{i=1}^{n} R_i$
50	24.4395	25.5605
100	33.5409	66.4591
150	37.9966	112.0034
200	40.4535	159.5465
250	42.1512	207.8488
300	43.3801	256.6199
350	43.8460	306.1540
400	44.2076	355.7924
450	44.2787	405.7213
500	44.0990	455.9010
550	44.1838	505.8162
600	44.2321	555.7679

APPENDIX D – CHAPTER 5'S ALGORITHMS AND RESULTS

Computational Complexity Analysis

Algorithm 5.1

ALGORITHM 5.1. Calculate the validity of the root agent's dependence relationship	
Function : evaluateDepRelationship()	
Input : agt_{root}, φ_{root}	
Output : *validity_of_relationship*	
validity = true	
$agt_{crt} = agt_{root}$	
WHILE have_child(agt_{crt}) **DO**	O(E)
IF have_child(agt_{crt}) **THEN**	
IF get_history(agt_{crt}) IS *valid* **THEN**	O(1)
dep = dentify_relationship(agt_{crt})	O(1)
IF dep IS dep_{or} **THEN**	
or_dependence()	O(V)
ELSE IF dep IS dep_{and} **THEN**	
and_dependence()	O(E^V)
ELSE IF dep IS dep_{1to1} **THEN**	O(1)
sing_dependence()	
ELSE	O(1)
terminate(agt_{crt})	
agt_{crt} = stack_remove()	O(1)
update(*parent_agent, weight*)	O(1)
ELSE	
IF check_validity() IS *invalid* **THEN**	
terminate()	O(1)
ELSE	
update(*parent_agent, weight*)	
update_information()	O(1)
ENDWHILE	
TOTAL COMPLEXITY	O(E^v)

Algorithm 5.2

ALGORITHM 5.2. Evaluate presence of agent agt_{crt} in history	
Function : get_history()	
Input : agt_{crt}	
Output : *validity*	
IF agt_{crt} IS *valid* **THEN**	
IF agt_{crt} IS NOT *visited* **THEN**	
stack_add(agt_{crt})	O(1)
validity = true	O(1)
ELSE	
stack_remove()	O(1)
ELSE	
stack_remove()	O(1)
RETURN *validity*	
TOTAL COMPLEXTY	O(1)

Algorithm 5.3

ALGORITHM 5.3. Evaluate the dep_{1to1} relationship	
Function : sing_dependence()	
Input : agt_{crt}	
Output : *validity*	
agt_{crt} = *agent*	
update(*current_capacity*, *value*(agt_{crt} , c))	O(1)
IF $c \leq \varphi$ **THEN**	O(1)
dep_{1to1} = valid	O(1)
ELSE	
dep_{1to1} = invalid	O(1)
RETURN *validity*	
TOTAL COMPLEXITY	O(1)

Algorithm 5.4

ALGORITHM 5.4. Evaluate the dep_{or} relationship	
Function : or_dependence()	
Input : agt_{crt}	
Output : *validity*	
FOR *agent* IN GET_LIST(dep_{or} , agt_{crt}) **DO**	O(V)
agt_{crt} = *agent*	O(1)
update(*current_capacity*, *value*(agt_{crt} , c))	O(1)
ENDFOR	
IF $c \leq \varphi$ **THEN**	O(1)
dep_{or} = valid	
ELSE	
dep_{or} = invalid	O(1)
RETURN *validity*	
TOTAL COMPLEXTY	O(V)

Algorithm 5.5

ALGORITHM 5.5. Evaluate the dep_{and} relationship	
Function : and_dependence()	
Input : agt_{crt}	
Output : *validity*	
dep_{and} = valid	O(1)
FOR *agent* IN GET_LIST(*and*, agt_{crt}) **AND** dep_{and} IS valid **DO**	O(E^v)
update_best_offer(*agent*)	
agt_{crt} = *agent*	O(1)
update(*current_capacity*, *value*(agt_{crt} , c))	O(1)
IF $c \leq \varphi$	O(1)
THEN	
dep_{and} = valid	O(1)
ELSE	
dep_{and} = invalid	O(1)
RETURN *validity*	
ENDFOR	
TOTAL COMPLEXTY	O(E^v)

Experimental Results

Experiment 5.1

n	$\sum_{j=1}^{n} \sigma_j$	$\sum_{j=1}^{n} v_j$	μ
10	338.186	345.58	30.0
20	659.776	601.65	86.0
30	820.010	935.16	148.6
40	867.568	940.694	260.2
50	1354.858	1385.81	286.0
60	1192.598	1260.21	595.6
70	1454.298	1663.846	742.4
80	1635.584	1602.134	773.6
90	1353.068	1288.322	962.0
100	1504.064	1560.772	1143.2

Experiment 5.2

	AND-Dependence, dep_{and}	***OR-Dependence,*** dep_{or}	***SINGLETON Dependence,*** dep_{1to1}	***Mixed Dependence,*** dep_{mix}
$\sum_{j=1}^{n} \sigma_j$	1022.221	389.119	1007.832	627.32
$\sum_{j=1}^{n} v_j$	1029.587	443.286	1006.613	662.253
μ	42.0	94.1	42.0	84.5

APPENDIX E – CHAPTER 6'S ALGORITHMS AND RESULTS

Computational Complexity Analysis

Algorithm 6.1

ALGORITHM 6.1. Agent agt_{join} joins a macroscopic based coalition.	
1. Searches for a compatible coalition where both have common goals through agent agt_{reg}.	O(1)
2. Retrieves the LOC from agent agt_{reg} and decides on the targeted coalition.	O(1)
3. Send the JCR to targeted coalition through agent agt_{reg}.	O($\Re$)
4. Waits for the reply from agent agt_{reg} regarding the voting session result.	O(1)
5. If agent agt_{join} did not receive agent agt_{reg} within the interval or received acknowledgement of JCR's failure, it will search alternate compatible coalitions by repeating Step 1. Otherwise, agent agt_{join} will proceed to Step 6.	O(1)
6. Play NIPD games with the targeted coalition by sending decision(s) to agent agt_{reg}.	O($n \times d$)
7. If $\sum_{h=0}^{n} T_{\alpha h} < T_{\omega}$, agent agt_{join} will revise its belief and repeat Step 1. Else, it will wait for agt_{reg} announcement to join the coalition.	O(1)
8. Officially joins the coalition and wait for tasks scheduling.	O(1)
TOTAL COMPLEXITY	$O\begin{pmatrix}\Re + \\ (n \times d)\end{pmatrix}$

Algorithm 6.2

ALGORITHM 6.2. Agent agt_{reg} handling the JCR.	
1. Receives request of joining coalition from agent agt_{join} with the goal g_{join}.	O(1)
2. Evaluates agent agt_{join} and ensures it is not engaged in any recent JCR activities (prevents spamming).	O($\Im$)
3. Searches the lists of compatible coalition with the goal g_{join} and return LOC to agent agt_{join}.	O($\Im\log(\Im)$)
4. Receives JCR from agent agt_{join} and the targeted coalition.	O(1)
5. The coalition z will be labelled "busy" for certain interval to prevent spamming.	O(1)
6. Organizes the voting section among the coalition members.	O(n)
7. If the voting section for the JCR passes, acknowledges agent agt_{join} and coalition members $Agt_x \in z$ before proceeds to the Step 7. Otherwise, acknowledge agent agt_{join} to terminate its JCR proposal.	O(1)
8. Conducts NIPD games monitoring between agt_{join} and $agt_x \in z$ through agent agt_{reg}.	O($2\times(n\times d)$)
9. Records the NIPD game results in H and $\overline{H}$. If $\overline{H} > H$, agt_{join}'s $T_\alpha - 3$. If $H > \overline{H}$, agt_{join}'s $T_\alpha + 1$.	O(1)
10. Announces the PD games result to agent agt_{join} and coalition.	O($\Im$)
11. Updates database if agent agt_{join} managed to win majority game.	O(1)
TOTAL COMPLEXITY	$O\begin{pmatrix} d+n+ \\ \Im\log(\Im) \end{pmatrix}$

Algorithm 6.3

ALGORITHM 6.3. Agent $agt_x \in z$ handling the JCR.	
1. Receives notification from agent agt_{reg} about agent agt_{join}'s JCR.	O(1)
2. Evaluates agent agt_{join}'s JCR.	O(1)
3. Votes by sending the decision to agent agt_{reg}.	O(1)
4. Waits for agent agt_{reg}'s acknowledgement of the voting result.	O(1)
5. If majority votes agree on agent agt_{join}'s JCR, proceed to Step 6. Otherwise, terminate agent agt_{join}'s JCR and continue tasks.	O(d)
6. Sends decision to agent agt_{reg} to play NIPD games with agt_{join}.	O(1)
7. Waits for agent agt_{reg}'s announcement of the NIPD game result.	O(1)
8. If NIPD games show agent agt_{join} won, wait for coalition reformation. Else, the coalition agents will continue with their current tasks.	O(NP)
TOTAL COMPLEXITY	O(NP)

Experimental Results

Experiment 6.1

P_{s1}	d	μ	E_{μ}	H	$\overline{H}$	sc	$\aleph$	T_{sim}
0.00	170849	97.6%	2.37%	959	41	460407	2.69	123.81
0.10	146825	83.9%	16.10%	689	311	344346	2.35	105.19
0.20	112469	64.3%	35.73%	183	817	226305	2.01	80.37
0.30	98651	56.4%	43.63%	20	980	186987	1.90	70.49
0.40	80542	46.0%	53.98%	0	1000	141423	1.76	57.87
0.50	80143	45.8%	54.20%	0	1000	140580	1.75	57.79
0.60	80395	45.9%	54.06%	0	1000	142317	1.77	57.95
0.70	66367	37.9%	62.08%	0	1000	107451	1.62	47.56
0.80	48468	27.7%	72.30%	0	1000	62280	1.28	34.97
0.90	39172	22.4%	77.62%	0	1000	41355	1.06	28.56
1.00	21000	12.0%	88.00%	0	1000	0	0.00	15.66

Experiment 6.2

Q_{join}	d	μ	E_{μ}	H	$\overline{H}$	sc	$\aleph$	T_{sim}
Desperate	117488	67.1%	32.86%	234	766	237279	2.0196	24075.23
Grumpy	36582	20.9%	79.10%	0	1000	89682	2.4515	24111.48
Imitator	121940	69.7%	30.32%	256	744	287256	2.3557	24229.32
Avenger	121240	69.3%	30.72%	239	761	285023	2.3509	24346.7
RandomGuy	65513	37.4%	62.56%	12	988	152240	2.3238	24408.93

APPENDIX F – CHAPTER 7'S ALGORITHMS AND RESULTS

Computational Complexity Analysis

Algorithm 7.1

ALGORITHM 7.1. Agent agt_{join}'s perspective in proposing JCR to the targeted coalition	
1. Requests the LOC from agent agt_{reg} that are compatible with its goals.	O(1)
2. Retrieves the LOC from agent agt_{reg}.	O(1)
3. Searches the highest compatible coalition from the LOC.	O($\Re$)
4. Proposes the JCR and the targeted coalition to the agent agt_{reg}.	O(1)
5. Waits for agent agt_{reg}'s respond for a certain interval.	O(1)
6. If agent agt_{join} did not receive agent agt_{reg}'s respond within the given interval or acknowledgement of JCR's failure, it will search for other compatible coalition and repeat Step 1. Otherwise, the agent agt_{join} will proceed to Step 7.	O(1)
7. Waits for the WVS's result through agent agt_{reg}.	O(1)
8. If $\sum_{x=1}^{n} \Theta_{no} \geq \Omega$, it will revise its belief and repeats Step 1. Else, it will wait for agt_{reg} announcement and waits for coalition reformation.	O(1)
TOTAL COMPLEXITY	O($\Re$)

Algorithm 7.2

ALGORITHM 7.2. Agent agt_{reg} handles agent agt_{join}'s JCR	
1. Receives the requests from agent agt_{join} with the goal g_{join}.	O(1) O($\Im$)
2. Checks agent agt_{join}'s status and ensure it is not engaged in any recent coalition activities.	$O\begin{pmatrix}\Re\times \\ \Im\log(\Im)+\Im\end{pmatrix}$
3. Searches and generate LOC for agent agt_{join}.	O(1)
4. Receives agent agt_{join}'s JCR and coalition of its choice.	O(1)
5. Forwards agent agt_{join}'s JCR to the targeted coalition will be labelled as "busy"	O(1)
6. Waits for agent agt_{join}'s evaluation result. If the evaluation result shows agt_{lead} approve the agt_{join}'s JCR, notify agent agt_{join} and proceed to Step 7. Otherwise, acknowledge agt_{join} about the rejection of JCR.	O(1)
7. Notifies agents to evaluate and organize the WVS.	
8. If $\sum_{x=1}^{n}\Theta_{yes}\geq\Omega$ or $\sum_{x=1}^{n}\Theta_{no}\geq\Omega$, terminate WVS. Otherwise, continue WVS until terminating condition fulfilled.	O(n)
9. Announces the WVS's results to agent agt_{join} and agents Agt_x.	O(1)
10. If $\sum_{x=1}^{n}\Theta_{yes}\geq(\sum_{x=1}^{n}\Theta_x\times\Omega)$, the agt_{reg} will inform the agt_{join} to join the coalition. Otherwise, it will inform agt_{join} to join an alternate coalition.	O(1) O(1)
11. Updates the databases if agent agt_{join} joined the coalition.	
TOTAL COMPLEXITY	$O(\Re\times\Im\log(\Im))$

Algorithm 7.3

ALGORITHM 7.3. Coalition representative agt_{lead}'s perspective of handling JCM.	
1. Receives the notification from agt_{reg} about agent agt_{join}'s JCR and notify coalition members $Agt_x \in z$.	O(1)
2. The agent agt_{lead} evaluates agent agt_{join}'s JCR.	O($\Re$)
3. If agent agt_{lead} has accepted agent agt_{join}'s JCR, it will proceed to Step 4. Otherwise, agt_{lead} will reject agt_{join}'s JCR and resume its operation.	O(1)
4. Requests a WVS with the $Agt_x \in z$ through agent agt_{reg}.	O(1)
5. Sends its vote to agent agt_{reg}.	
6. Waits for agent agt_{reg}'s announcement on the WVS's result.	O(1)
7. If $\sum_{x=1}^{n} \Theta_{yes} \geq \Omega$, agent agt_{lead} will reform the coalition by including agent agt_{join} and the coalition will resume its operation. Otherwise, its operation will be resuming without coalition reformation.	O(1) O(1)
TOTAL COMPLEXITY	O($\Re$)

Algorithm 7.4

ALGORITHM 7.4. Coalition member agt_x's perspective of handling JCM.	
1. Receives the notification from agent agt_{lead} about agent agt_{join}'s JCR.	O(1)
2. Evaluates agent agt_{join}'s JCR then allocate by preference to the Θ_{no} and Θ_{yes}.	O($\Re$)
3. Resumes existing tasks until received WVS notification from agent agt_{reg}.	O(1)
4. Suspends current tasks and send Θ_{yes} and Θ_{no} to agent agt_{reg}.	O(1)
5. Resumes existing tasks until receiving the WVS's result notification from agent agt_{reg}.	O(1)
6. If WVS passed, waits for the coalition reformation order from agent agt_{lead}. Otherwise, resume the existing operation.	O(1)
TOTAL COMPLEXITY	O($\Re$)

Experimental Results

Experiment 7.1

n	T_{sim}	$\sum_{x=1}^{n}\Theta_{yes}$	$\sum_{x=1}^{n}\Theta_{no}$	ϖ_{IAV}	ϖ_{IDV}	ϖ_{CAV}
50	14.796	298144	304430	43.43%	51.53%	5.04%
100	30.153	347216	363070	38.59%	40.36%	21.05%
150	46.510	457397	481997	25.42%	26.79%	47.79%
200	63.179	586724	609853	19.56%	20.33%	60.11%
250	80.511	715850	743583	15.91%	16.53%	67.56%
300	98.240	847315	873911	13.45%	13.87%	72.67%
350	115.688	971275	1001275	11.56%	11.92%	76.51%
400	132.911	1104983	1138432	10.23%	10.54%	79.22%
450	150.122	1233589	1270672	9.14%	9.41%	81.45%
500	167.521	1362784	1410867	8.26%	8.55%	83.19%

Experiment 7.2

$CR_{\Theta yes}$	T_{sim}	$\sum_{x=1}^{n}\Theta_{yes}$	$\sum_{x=1}^{n}\Theta_{no}$	ϖ_{IAV}	ϖ_{IDV}	ϖ_{CAV}
0.00	15.374	0	892893	0.00%	71.42%	28.58%
0.10	31.343	88748	1778191	3.55%	71.12%	25.33%
0.20	47.046	294719	2643164	7.86%	70.47%	21.67%
0.30	62.508	628460	3455665	12.57%	69.10%	18.33%
0.40	78.291	1135185	4240408	18.16%	67.84%	14.00%
0.50	93.698	1881584	5001861	25.09%	66.69%	8.23%
0.60	109.074	2663419	5539984	30.44%	63.31%	6.25%
0.70	124.420	3469238	5900026	34.69%	59.00%	6.31%
0.80	139.830	4327188	6129200	38.46%	54.48%	7.05%
0.90	155.218	5209486	6239119	41.68%	49.91%	8.41%
1.00	170.575	6099465	6257339	44.36%	45.51%	10.13%

Experiment 7.3

$(\Omega \times 100)$	T_{sim}	$\sum_{x=1}^{n} \Theta_{yes}$	$\sum_{x=1}^{n} \Theta_{no}$	ϖ_{IAV}	ϖ_{IDV}	ϖ_{CAV}
10	8.0	13952634	14393955	9.30	9.59	81.11
15	8.7	35053380	36132264	11.69	12.05	76.27
20	9.1	63299163	65302969	14.07	14.51	71.42
25	9.9	98766197	101945120	16.46	16.99	66.55
30	10.9	141495509	145982417	18.87	19.46	61.67
35	17.1	191428723	197446884	21.27	21.94	56.79
40	29.9	248599973	256349501	23.68	24.41	51.91
45	30.1	313048080	322792714	26.09	26.90	47.01
50	30.5	385995909	397355743	28.59	29.43	41.98
55	30.8	460442320	472938201	30.69	31.53	37.78
60	31.0	534889379	548526832	32.42	33.24	34.34
65	31.3	609296642	624100753	33.85	34.67	31.48
70	31.6	683758640	699663382	35.06	35.88	29.06
75	31.8	758200638	775208917	36.10	36.91	26.98
80	32.0	832671139	850777769	40.72	41.61	17.67
85	32.3	907108260	926328298	47.87	48.89	3.24
90	32.6	981556925	1001908258	48.89	49.83	1.28