2018

Computational Approaches for Coordinating Multiple Concurrent Negotiations

Lei Niu
University of Wollongong

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author.

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation


Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Computational Approaches for Coordinating Multiple Concurrent Negotiations

Lei Niu

Supervisor:
Dr. Fenghui Ren
Co-supervisor:
Prof. Minjie Zhang

This thesis is presented as required for the conferral of the degree:

Doctor of Philosophy

The University of Wollongong
School of Computing and Information Technology

April, 2018
Declaration

I, Lei Niu, declare that this thesis submitted in fulfilment of the requirements for the conferral of the degree Doctor of Philosophy, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Lei Niu
September 5, 2018
Abstract

Agent negotiation is a method of compromise between intelligent agents to reach a bi-beneficial agreement over negotiated issues. In the last two decades, agent negotiation has led to great achievements. However, most of the researchers focused on single negotiation, where only one negotiation is involved. Little work has been done on addressing challenging research issues in multiple negotiations, where multiple interdependent negotiations are involved. Multiple Concurrent Negotiations (MCN) indicate one of the multiple negotiations scenarios, where multiple negotiations with interdependent issues are concurrently conducted. This thesis aims at: (1) investigating some significant challenging research issues in MCN, and (2) developing novel solutions for the reported challenges in order to effectively handling MCN.

This thesis tries to solve five challenges in MCN, which are: (1) handling the issue interdependency across MCN, (2) handling the concurrency in MCN, (3) developing protocols for MCN in a dynamic changing negotiation environment, (4) designing negotiation procedures for MCN, and (5) handling social factors of MCN in an agent society.

To solve the five challenges, the contributions of this thesis consist of the following four aspects.

1. An MCN model is proposed, where issue interdependency is represented by a graph-based model and a Colored Petri Net-based negotiation protocol is presented to handle the concurrency in MCN.

2. An MCN protocol is proposed to handle the changes in a dynamic changing negotiation environment, where the number of negotiations can be changed during the negotiations. In the proposed protocol, MCN is represented by a Colored Petri Net-based model. Through applying a serial of adaptive manipulations to the Colored Petri Net-based model for MCN, the dynamic changes of negotiations can be effectively handled in a dynamic changing negotiation environment.

3. Three negotiation procedures for MCN are proposed in this thesis, which are the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. The experimental results reveal that:
(1) the successive negotiation procedure achieves the best performance in terms of negotiation efficiency, (2) the concurrent negotiation procedure achieves the best performance in terms of negotiation effectiveness, and (3) the clustered negotiation procedure provides a well-balanced solution between negotiation effectiveness and efficiency.

4. A trust-based approach is proposed to handle social factors of MCN in an agent society. In the proposed approach, agents take their opponents’ reputations into consideration in the decision-making process on proposing offers. The experimental result shows that the proposed trust-based approach can effectively and efficiently handle social factors of MCN in an agent society.

In summary, this thesis investigates the challenging research issues in MCN and proposes several solutions to these challenges. Experimental results reveal the effectiveness and efficiency of the proposed approaches in this thesis.
Acknowledgments

In the past three and a half years Ph.D. study, I could not complete my research without the help and support I received from many people.

Foremost, I am indebted to my supervisors, Dr. Fenghui Ren and Prof. Minjie Zhang, for their patience, encouragement, immense knowledge, and for their continuous support and guidance for not only my Ph.D. study but also my daily life. Without their help, this thesis would not be completed.

My thanks are extended to my colleagues and friends at University of Wollongong, Xing Su, Yan Kong, Xishun Wang, Jihang Zhang, Yuchen Wang and Yan Zhao. I also would like to thank Weihua Li from Auckland University of Technology, for his research discussions.

Studying abroad is a tedious journey. I would like to express my thanks to my friends, Erik Brettingen Johansen, Jacob Terry Rath, Jerome Clinton Justin, and Alexander Richard Nagle, who gave me lots of happiness and fun and helped me in improving English. I would like to give special thanks to Erik Brettingen Johansen, who also gave me kind help in proofreading this thesis and in understanding western culture.

I would like to express my deepest gratitude to my parents, Quanxiang Niu and Qiaoyun Zhang, for their encouragement, support, understanding and love.

In addition, I gratefully acknowledge the scholarships provided by University of Wollongong, namely the University Postgraduate Award and the International Postgraduate Tuition Award.

Finally, thanks to all the anonymous reviewers of my research papers and my thesis, and all my other friends and relatives who have supported me.
Publications

The following is a list of my research papers, which have already been published (accepted) or are under working during my PhD study that ends with the completion of this thesis.

Refereed Journal Articles:


Refereed Conference Papers:


Papers under Working:

6. Lei Niu, Fenghui Ren and Minjie Zhang, “A Trust-Based Mechanism for Handling Multiple Interdependent Negotiations in an Agent Society”, will be submitted to Artificial Intelligence Journal.
Contents

Abstract iii

1 Introduction 1
  1.1 Background of Agent Negotiation 1
      1.1.1 Main Elements in Agent Negotiation 2
      1.1.2 Classifications of Agent Negotiation 3
      1.1.3 A Hierarchical View of Agent Negotiation 4
  1.2 Multiple Concurrent Negotiations 5
  1.3 Research Challenges in MCN 7
  1.4 Major Research Issues in MCN 9
  1.5 Objectives and Contributions of This Thesis 10
      1.5.1 Objectives of This Thesis 10
      1.5.2 Contributions of This Thesis 11
  1.6 The Structure of This Thesis 13

2 Literature Review 14
  2.1 Issue Interdependency 14
  2.2 Concurrent Negotiation 18
  2.3 Dynamic Changing Negotiation Environments 21
  2.4 Negotiation Procedures 24
  2.5 Reputation-Based Systems 26
  2.6 Summary 28

3 A Multiple Concurrent Negotiations Model 30
  3.1 Problem Formulation 30
      3.1.1 Definitions 30
      3.1.2 A Classification of MCN 32
  3.2 A Graph-Based Modelling for MCN 33
      3.2.1 A Graph-Based Representation of MCN 33
      3.2.2 A Logic-Based Representation of MCN 33
      3.2.3 Utility Calculation for MCN 35
List of Tables

3.1 Negotiation parameters generated randomly in a complete success scenario ........................................ 52
3.2 Negotiation parameters generated randomly in a partial success scenario 54
3.3 Negotiation parameters generated randomly in a no success scenario . 57

4.1 Parameters in single negotiations ................................................................. 67
5.1 Parameters in single negotiations ................................................................. 84
5.2 Experimental settings for Case (a) .............................................................. 85
5.3 Experimental settings for Case (b) .............................................................. 85
5.4 Parameters in issue interdependencies ...................................................... 85

6.1 Parameters in every single negotiation ....................................................... 101
6.2 Experimental settings for MCN in Scenario (a) ................................. 102
6.3 Experimental settings for MCN in Scenario (b) ................................. 102
6.4 Parameters in issue interdependencies ...................................................... 102
List of Figures

1.1 A three-level hierarchical view of agent negotiation ................. 4
1.2 An example of issue interdependency in MCN ......................... 6

2.1 An example of issue interdependency across multiple negotiations . 18
2.2 System architecture of concurrent negotiation [NJ03b] ................. 19

3.1 Graph-based representations of five types of MCN .................... 34
3.2 The CPN-based representation of a single MCN ......................... 37
3.3 The CPN-based representation of a sequential MCN .................... 37
3.4 The CPN-based representation of a synchronized MCN ................ 38
3.5 The CPN-based representation of a merging MCN ....................... 39
3.6 The CPN-based representation of a hybrid MCN ......................... 40
3.7 MCN N’s graph-based representation ..................................... 47
3.8 The percentage of multi-negotiation outcomes .......................... 47
3.9 Final utilities achieved from single negotiations .................... 49
3.10 Final utilities achieved from MNRs and MCN .......................... 50
3.11 Round intervals of single negotiations .................................. 51
3.12 Number of rounds and average deadline of agents in single negotiations 51
3.13 Utilities achieved from single negotiations in a complete success scenario 53
3.14 Utilities achieved from MNRs in a complete success scenario ........ 54
3.15 Utilities achieved from single negotiations in a partial success scenario 55
3.16 Utilities achieved from MNRs in a partial success scenario ........... 56
3.17 Utilities achieved from single negotiations in a no success scenario . 57
3.18 Utilities achieved from MNRs in a no success scenario ................. 58

4.1 A graph-based representation updating example of changing negotiations .............................................. 61
4.2 The CPN-based representation of an MNR ............................... 63
4.3 An example of the graph-based representation updating for MCN .... 64
4.4 An example of the CPN-based representation updating for the MCN 64
4.5 The basic MCN ..................................................................... 67
4.6 Possible positions of changing negotiations ............................ 68
LIST OF FIGURES

4.7 Percentage of an agent’s MNG in the scenario of adding negotiations 69
4.8 Overall utility from negotiations in the scenario of adding negotiations 70
4.9 Percentage of an agent’s MNG in the scenario of removing negotiations 71
4.10 Overall utility from negotiations in the scenario of removing negotiations 71
4.11 Percentage of an agent’s MNG in the scenario of simultaneously adding and removing negotiations 72
4.12 Overall utility from negotiations in the scenario of simultaneously adding and removing negotiations 73

5.1 Edgeworth-Bowley box for an agent’s strategy affected by issue interdependencies across MCN 78
5.2 An example of issue interdependency across MCN 81
5.3 Agent’s overall utility in Case (a) 86
5.4 Agent’s overall utility in Case (b) 87
5.5 Success rate of negotiations in Case (a) 88
5.6 Success rate of negotiations in Case (b) 88
5.7 Number of negotiation rounds in Case (a) 89
5.8 Number of negotiation rounds in Case (b) 90

6.1 An example of a negotiation society’s hierarchical graph-based representation 93
6.2 Edgeworth-Bowley box for an agent’s strategy affected by opponent’s trust value and issue interdependencies 98
6.3 Social welfare in Scenario (a) 103
6.4 Social welfare in Scenario (b) 104
6.5 Success rate of negotiations in Scenario (a) 105
6.6 Success rate of negotiations in Scenario (b) 105
6.7 The number of negotiation rounds in Scenario (a) 106
6.8 The number of negotiation rounds in Scenario (b) 107
Chapter 1

Introduction

1.1 Background of Agent Negotiation

An autonomous agent is a type of computer system situated in an environment, and it can take flexible autonomous actions to meet the agent’s original design objectives [DKV11].

Agent negotiation is a process between two or more agents who intend to achieve a negotiation agreement over one or more negotiation issues [Kra01, Rai82]. An example of a negotiation is that two agents are bargaining in an electronic marketplace over the price of an item. The negotiation will proceed based on a predefined negotiation protocol acknowledged by both negotiating agents, i.e., the two agents exchange offers alternatively. During the process of a negotiation, the negotiation is conducted through making concessions to reach an agreement. A negotiation usually starts from one agent with an initial offer, and the other agent responds by either accepting the offer, rejecting the offer or quitting from the negotiation [Rub82]. The negotiation is usually processed in several rounds, and the offer, made by each agent in each round, is calculated based on the agents’ individual negotiation strategies. An agent’s private information, such as the negotiation deadlines, the initial offers and the reserved offers, can affect its negotiation strategy. Finally, if any agent’s offer is accepted by the other agent, the negotiation succeeds. Otherwise, the negotiation fails.

Agent negotiation is a very important topic in the research area of agent and multi-agent systems. Agent negotiation has attracted great attention in last few decades, and it is widely employed in broad domains, such as e-commerce, resource allocation, cloud computing, etc.
1.1.1 Main Elements in Agent Negotiation

Generally speaking, there are mainly three elements, significantly impacting agent negotiation, which are the negotiation protocol, the negotiation procedure, and the negotiation strategy.

1. Negotiation protocol

A negotiation protocol defines a set of rules, which dictates the conduct between agents to achieve an outcome. The alternating offer protocol [Rub82] is one of the most commonly employed negotiation protocols. In the alternating offer protocol, either of the two involved agents can start the negotiation by proposing an initial offer. The opponent agent will reply by accepting the offer, rejecting the offer or quitting from the negotiation. In the case of accepting the offer, the negotiation succeeds in making an agreement; in the case of rejecting the offer, the opponent proposes a counter-offer, and waits for the response from the agent; in the case of quitting from the negotiation, the negotiation fails in making an agreement.

2. Negotiation procedure

A negotiation procedure specifies how negotiation issues are processed in multi-issue negotiation [FWJ04]. Generally speaking, three general negotiation procedures [FWJ06a, FWJ06b] are widely used in multi-agent research, which are the simultaneous procedure, the sequential procedure, and the package deal procedure. The simultaneous procedure processes all issues simultaneously, but independently of each other. The sequential procedure processes all issues one after another. The package deal procedure processes all issues together in a bundle.

3. Negotiation strategy

A negotiation strategy indicates a pre-determined approach to generate negotiation actions in order to achieve a negotiation agreement between agents. For different negotiation domains, agents will use different negotiation strategies to efficiently achieve their negotiation goals. In the competitive negotiation domain, agents aim at maximizing their individual profits, while in the cooperative negotiation domain, agents will cooperate with each other to maximize the overall profit of all agents. In a more complex domain, agents may apply a combined negotiation strategy to balance the profit between themselves and the overall domain.
1.1.2 Classifications of Agent Negotiation

Agent negotiation can be classified into different categories by considering different criteria as follows.

1. The number of negotiation issues

Considering the number of negotiation issues, agent negotiation can be classified into single-issue negotiation and multi-issue negotiation [FWJ04]. Single-issue negotiation indicates that only one issue is under negotiation, while multi-issue negotiation indicates more than one issue is under negotiation.

2. The number of negotiation participators

Based on the number of negotiation participators, agent negotiation can be classified into bilateral negotiation and multilateral negotiation [LGS06]. In bilateral negotiation, an agent negotiates with exactly one opponent to achieve an agreement, while in multilateral negotiation, an agent negotiates with more than one opponent.

3. The status of negotiation environments

Based on the status of negotiation environments, agent negotiation can be classified into static negotiation and dynamic negotiation [RZS09]. In static negotiation, there are no agents entering into or leaving from the environment during the negotiation, which means that agents make offers and decisions in the negotiation only based on a static negotiation situation. In dynamic negotiation, agents may enter into or leave from the environment during the negotiation. Therefore, the negotiation environment in dynamic negotiation is open, and agents need to make negotiation offers and decisions considering changes happened during the negotiation process.

4. The number of negotiations

Regarding the number of negotiations, agent negotiation can be classified into single negotiation and multiple negotiations. From current literature, most of the previous classifications focus on single negotiation. However, the real-world negotiation scenarios could be much more complicated, where (1) multiple negotiations with different negotiation goals might be considered, and (2) these negotiation goals could be somehow interdependent with each other.
1.1.3 A Hierarchical View of Agent Negotiation

A three-level hierarchical view of agent negotiation is displayed in Figure 1.1 based on the classifications of agent negotiation presented in Section 1.1.2. The view includes three levels, which are the bilateral negotiation level, the multilateral negotiation level, and the multi-negotiation level. The bilateral negotiation level covers bilateral negotiation, where a single-issue or a multi-issue negotiation with only one negotiation goal is conducted between two agents; the multilateral negotiation level covers multilateral negotiation, where the negotiation with only one goal is conducted between multiple agents; and the multi-negotiation level covers multiple negotiations with different negotiation goals.

A vast majority of the research done on single negotiation has led to great achievements in recent years, but very little work has been done on multiple negotiations.

Multi-negotiation scenarios are very common in the real world. For instance,
through using the Internet, a travel agent can interact with other agents for a trip package with a budget and time requirement, which may include a flight, accommodation and some tourist activities. Typically, the agent conducts multiple negotiations with corresponding agents for flights, accommodations and activities, respectively, and these negotiations are interdependent when considering the factors of budget, sequence and time. During the negotiations, the overall goal of the agent is to achieve a series of negotiation agreements with the lowest price and a suitable schedule to satisfy the budget and the time requirement. In order to achieve such a goal, an appropriate approach for multiple negotiations must be used to conduct these negotiations in an open environment while considering constraints and interdependencies of these multiple negotiations. However, the research on multiple negotiations is still challenging due to the limitations in current research. In order to fill this gap, this thesis focuses on the research of multiple negotiations.

1.2 Multiple Concurrent Negotiations

Multiple Concurrent Negotiations (MCN) indicate one of the scenarios in multiple negotiations. In MCN, an agent negotiates with other opponents through concurrently conducting multiple negotiations to achieve an overall goal. For every single negotiation in MCN, the agent has an independent negotiation goal, and issues in a single negotiation could be dependent on the issues in other negotiations.

Therefore, the main features of MCN are: (1) the concurrency in MCN and (2) the issue interdependency across MCN.

The detailed explanations on the two main features of MCN are described as follows.

1. Concurrency in MCN

During the process of MCN, issues within different negotiations could be interdependent with each other. Therefore, through concurrently processing these multiple negotiations, the agent could simultaneously consider all offers received from different negotiations in deciding the agent’s negotiation actions to optimise the negotiation outcome. Obviously, concurrently handling MCN is one of the challenging problems in MCN research.

2. Issue interdependency across MCN

In agent negotiation, negotiation issues may have different relationships with each other. For multi-issue negotiation, the issues in the negotiation could be independent or interdependent. For the case of independent issues, an agent’s profit gained from an issue only depends on the issue itself. For the case of
interdependent issues, an agent’s profit gained from an issue is also affected by other interdependent issues [BY97, KFSBY03a, FWJ06a].

For MCN, issues in different negotiations could be independent or interdependent. For the case of independent issues, MCN can be viewed as multiple independent negotiations. For the case of interdependent issues, an agent’s profit achieved from a negotiation can also be impacted by other interdependent negotiations.

Usually, the interdependency in MCN is determined by the interdependencies of negotiation issues included in MCN.

Figure 1.2: An example of issue interdependency in MCN

Figure 1.2 shows an example of issue interdependency in MCN. The three big circles indicate three negotiations, i.e., $N_0$, $N_1$ and $N_2$; and the dots, squares and triangles in the big circles indicate negotiation issues, respectively. The overlapped shaded areas between negotiations indicate the interdependency between negotiation issues. In current literature, a large amount of work exclusively studies the issue interdependency within a particular multi-issue negotiation [FWJ06a, IKH06, KJ08, ARR13b], which is represented by the non-shaded areas in Figure 1.2. However, very little work considers the issue interdependency across MCN due to its complexity. The research on issue interdependency across MCN is also one of the focuses in this thesis.
1.3 Research Challenges in MCN

In recent years, research on single negotiation has led to great achievements, but very little work on multiple negotiations has been done. In order to cover the limitations of the current work, this thesis will focus on solving research problems in MCN. In this section, four research challenges in MCN are summarised as follows.

- **Challenge 1: MCN model design**

  Due to the limitations of the current literature, there are very few models for MCN. In order to design an appropriate model for MCN, the following four aspects need to be well considered.

  1. *Issue interdependency across MCN*

     Negotiations in MCN may be somehow related and could impact each other. Moreover, there might be different types of issue interdependencies in MCN. Therefore, issue interdependency across MCN is an important aspect in the MCN model design.

  2. *Concurrency in MCN*

     In MCN, an agent can concurrently exchange offers with its opponents in different negotiations, and it is unpredictable when the agent receives the offer from a specific negotiation. In each round of processing MCN, the agent may simultaneously receive multiple offers from different negotiations. These factors make it difficult for the agent to decide how to make an offer for every single negotiation while considering the current status of all negotiations and the issue interdependency across MCN. An inappropriate mechanism for handling the concurrency would probably result in the deadlock during the process of MCN. Therefore, handling the concurrency in MCN is another aspect, which should be considered in the MCN model design.

  3. *Dynamic changing negotiation environments*

     The dynamic changing negotiation environments in MCN indicate that the number of negotiations could be changed during the process of MCN. Any ongoing negotiation could be terminated, and new negotiations could be added at any time. As the changes in a negotiation environment may happen at any time, the unpredictable changes would alter the existing issue interdependency between the negotiations, which will cause difficulties for the agent to perform actions in MCN. Hence, the dynamic changing negotiation environment is the third aspect, which should be
considered in the MCN model design.

4. Social factors in the agent society

In a complicated negotiation environment, multiple agents involved into MCN will form an agent society, where a large number of agents interact with each other to achieve their negotiation goals. In an agent society, especially for long-term cooperations, agents would be more likely to keep stable and long-term cooperative relationships with those agents who have trustable reputations. Therefore, handling social factors of MCN in an agent society is the fourth aspect, which should be considered in the MCN model design.

To summarise, designing an MCN model while simultaneously considering the issue interdependency across MCN, the concurrency in MCN, dynamic changing negotiation environments, and agents’ reputations, is the first research challenge considered by this thesis.

• Challenge 2: MCN protocol design

In each round of a single negotiation, an agent only needs to wait for the response from its opponent in one negotiation, and then performs actions based on the opponent’s responses. However, during the process of MCN, an agent may receive multiple offers from different agents at the same time, and offers in MCN are frequently changing. Hence, an appropriate negotiation protocol is needed to effectively arrange the exchanges of offers in MCN, e.g., how to generate offers for every single negotiation while considering the issue interdependency across MCN and how to decide the order of sending the new generated offers to opponents in different negotiations. Moreover, in a dynamic changing negotiation environment, the unpredictable changes of negotiations could make the MCN protocol design very difficult. Also, an ineffective negotiation protocol for MCN would cause a significant reduction in the success rate of negotiations in MCN. Therefore, designing an effective negotiation protocol for MCN is the second research challenge considered by this thesis.

• Challenge 3: MCN procedure design

The ways of conducting multiple negotiations could be very different, which can vastly impact the outcome of MCN. Besides concurrently conducting negotiations in MCN, there may exist other negotiation procedures for processing MCN under different requirements in order to achieve good negotiation effectiveness and efficiency. Therefore, designing feasible MCN procedures while
considering negotiation effectiveness and efficiency is the third research challenge considered by this thesis.

- Challenge 4: MCN strategy design

In MCN, an agent has a negotiation goal for every single negotiation. These goals may be interdependent and impact with each other. With such interdependencies between negotiations, the agent may have to apply different negotiation strategies in each negotiation to optimise the negotiation outcome. During the process of MCN, an agent may also have to adaptively adjust its negotiation strategies due to unpredictable changes in a dynamic changing negotiation environment. Therefore, designing suitable negotiation strategies for optimising MCN outcomes is the fourth research challenge considered by this thesis.

1.4 Major Research Issues in MCN

Based on the research challenges identified in Section 1.3, this thesis focuses on investigating five major research issues, which are:

- **Issue 1: How to handle the issue interdependency across MCN?**

  In MCN, the issue interdependency across MCN may exist between negotiations and would impact the process of MCN, e.g., the negotiation procedure, the negotiation protocol, and the outcome of MCN. Hence, handling the issue interdependency across MCN is the first research issue addressed in this thesis.

- **Issue 2: How to handle the concurrency in MCN?**

  In MCN, multiple negotiations are conducted concurrently. Agents need to interact with each other, the number of possible sequences of processing MCN can be very large, and the outcome of MCN could be indeterminate. Therefore, effectively handling the concurrency in MCN is the second research issue addressed in this thesis.

- **Issue 3: How to handle MCN in a dynamic changing negotiation environment?**

  As discussed in Section 1.3, a dynamic changing negotiation environment in MCN indicates the changes of the number of negotiations. The unpredictable negotiation environment and the issue interdependency across MCN would enlarge the complexity of handling MCN. Therefore, effectively handling MCN
in a dynamic changing negotiation environment is the third research issue addressed in this thesis.

- **Issue 4: How to design negotiation procedures for MCN?**

During the process of MCN, complicated issue interdependencies and different negotiation procedures would vastly affect the process and the outcome of MCN. Regarding negotiation effectiveness and efficiency, different negotiation procedures would achieve different negotiation outcomes. Hence, proposing feasible negotiation procedures for MCN while considering negotiation effectiveness and efficiency is the fourth research issue addressed in this thesis.

- **Issue 5: How to handle social factors of MCN in an agent society?**

In order to keep stable and long-term cooperative relationships with trustable opponents in an agent society, the opponents’ social factors should be considered by an agent in the decision-making process during the negotiations. For instance, how to select an opponent in establishing long-term and stable cooperation relationships, and how to propose offers during the negotiations while considering the opponent’s social reputation. Therefore, effectively processing MCN while considering social factors in an agent society is the fifth research issue addressed in this thesis.

### 1.5 Objectives and Contributions of This Thesis

Based on the five research issues described in Section 1.4, four objectives of this thesis are introduced as follows.

#### 1.5.1 Objectives of This Thesis

- **Objective 1: To design an MCN model for handling the issue interdependency and the concurrency**

  The issue interdependency across MCN and the concurrency in MCN are two major research issues in the MCN model design. Therefore, the first objective of this thesis is to design an MCN model for handling the issue interdependency across MCN and the concurrency in MCN.

- **Objective 2: To propose a negotiation protocol for handling MCN in a dynamic changing negotiation environment**

  To investigate a dynamic changing negotiation environment in MCN, this thesis focuses on the negotiation protocol design for MCN while considering un-
predictable changes of MCN. The second objective of this thesis is to propose a negotiation protocol for conducting MCN in a dynamic changing negotiation environment.

- **Objective 3: To propose negotiation procedures for MCN**

Regarding negotiation effectiveness and efficiency, there may exist other negotiation procedures for MCN, which might achieve better performance than concurrently processing MCN in particular domains. Therefore, the third objective of this thesis is to design feasible negotiation procedures to process MCN by considering negotiation effectiveness and efficiency.

- **Objective 4: To propose an approach for handling social factors of MCN in an agent society**

In an agent society with a cooperative negotiation environment, agents would like to keep long-term cooperations with other agents through the negotiations. The fourth objective of this thesis is to propose a trust-based approach for handling social factors of MCN in an agent society.

### 1.5.2 Contributions of This Thesis

Focusing on the five major research issues and four objectives of this thesis, four contributions of this thesis are claimed as follows.

- **Contribution 1: An MCN model is proposed for concurrently handling multiple negotiations with issue interdependencies (to achieve Objective 1)**

A negotiation model for MCN is proposed in this thesis. The main contributions of the proposed model are that: (1) formal definitions related to MCN are introduced; (2) the issue interdependency across MCN is represented by a graph-based model; and (3) a Colored Petri Net-based negotiation protocol is proposed for handling the concurrency in MCN. The effectiveness of the proposed MCN model is evaluated through a set of experiments and case studies by considering different negotiation scenarios.

- **Contribution 2: An MCN protocol is proposed for handling MCN in a dynamic changing negotiation environment (to achieve Objective 2)**

An MCN protocol considering a dynamic changing negotiation environment is proposed in this thesis. In the proposed protocol, the issue interdependency is mathematically represented by a peer-to-peer graphic model. MCN is
represented by a Colored Petri Net-based model. Through applying a series of adaptive manipulations on the Colored Petri Net-based model for MCN, the dynamic changes of negotiations can be effectively handled in the dynamic changing negotiation environment. The effectiveness and the efficiency of the proposed protocol are evaluated through an experiment with comprehensive experimental settings, where all possible positions of adding and removing negotiations during the process of MCN are considered.

• Contribution 3: Three negotiation procedures for MCN are proposed for handling MCN in different ways (to achieve Objective 3)

Three negotiation procedures for MCN are proposed in this thesis, which are the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. In the concurrent negotiation procedure, an agent processes multiple negotiations concurrently. In the successive negotiation procedure, an agent processes multiple negotiations one after another. In the clustered negotiation procedure, all negotiations are firstly partitioned into some clusters, (i.e., disjoint subsets). Then, an agent processes negotiations in each cluster by using the concurrent negotiation procedure, and processes all clusters by using the successive negotiation procedure. The performances of the three proposed negotiation procedures are also analysed through an experiment. The experimental results show that: (1) the successive negotiation procedure achieves the best performance in terms of negotiation efficiency, (2) the concurrent negotiation procedure achieves the best performance in terms of negotiation effectiveness, and (3) the clustered negotiation procedure provides a well-balanced solution between negotiation effectiveness and efficiency.

• Contribution 4: A trust-based approach is proposed in handling social factors of MCN in an agent society (to achieve Objective 4)

A trust-based negotiation approach is proposed to handle social factors of MCN in an agent society. The main contributions of the proposed approach are that: (1) a general model for a negotiation society is proposed; and (2) a trust-based model is proposed for agents in conducting MCN with those agents in an agent society, where agents take their opponents’ reputations into consideration in the decision-making process on proposing offers during negotiations. The effectiveness and the efficiency of the proposed approach are evaluated through an experiment, and the experimental results show that the proposed trust-based approach can effectively and efficiently handle social factors of MCN in an agent society.
1.6 The Structure of This Thesis

This thesis starts with an overview of the background of agent negotiation and introduces the concept of MCN, research challenges in MCN, five major research issues in MCN, objectives and contributions of this thesis. The remaining chapters of this thesis are organised as follows.

Chapter 2 is a literature review of the current research on MCN. Research issues of issue interdependency, concurrent negotiation, dynamic changing negotiation environments, negotiation procedures, and reputation-based systems are discussed.

Chapter 3 introduces a negotiation model for MCN, where MCN is conducted concurrently by considering the issue interdependency.

Chapter 4 presents a negotiation protocol for MCN in a dynamic changing negotiation environment, where multiple interdependent negotiations could be added and removed during the MCN process.

Chapter 5 proposes three negotiation procedures for MCN, which are the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure.

Chapter 6 presents a trust-based approach for handling MCN in an agent society, where agents conduct MCN by considering their opponents’ reputations in an agent society.

Chapter 7 provides a conclusion of this thesis and indicates future work.
Chapter 2

Literature Review

In recent years, the research on agent negotiation has achieved great achievements, and a number of approaches have been proposed to handle different problems in single negotiation, such as negotiation environments [LGS06, AST+06], negotiation strategies [FSJ98, SGS09, SS12, HGA15], negotiation protocols [ZR91, PTJ03] and negotiation procedures [FWJ02, FWJ04]. Most of the current work focuses on single negotiation, and there is little work on addressing challenging problems in Multiple Concurrent Negotiations (MCN). However, with the rapid increase of the complexity of negotiation environments, the research on MCN requires a thoughtful study.

By considering both the single negotiation level and the multiple negotiations level, this chapter reviews the current work on the five major research issues described in Section 1.4. The outline of this chapter is as follows. Section 2.1 presents negotiation approaches on issue interdependency in single negotiation and issue interdependency across multiple negotiations. Section 2.2 introduces a detailed review of work on handling the concurrency in single negotiation and multiple negotiations. Section 2.3 reviews the current research on dynamic changing negotiation environments in single negotiation and multiple negotiations. Section 2.4 reviews current approaches on the procedures for both single negotiation and multiple negotiations, and Section 2.5 reviews approaches on reputation-based systems. Finally, Section 2.6 summarises this chapter.

2.1 Issue Interdependency

In this section, negotiation approaches on issue interdependency are reviewed. Regarding the hierarchical view of agent negotiation described in Section 1.1, issue interdependency could exist in single negotiation and across MCN.
• **Issue interdependency in single negotiation**

In single negotiation with multiple interdependent issues, an agent’s profit achieved from an issue does not only depend on that issue but also on other interdependent issues. It is difficult for the agent to make decisions on proposing offers for these interdependent issues, and the interdependency between issues would vastly affect the process of the negotiation. Interdependent issues result in the cases of linear utility functions and nonlinear utility functions in calculating agents’ profits. In the case of linear utility functions, negotiation issues are related to each other, and the relationship between the issues is linear, while the case of independent issues does not require such a restriction. Handling the case of nonlinear utility functions is more challenging as it is hard to find optima because of the nonlinearity of agents’ utility functions [BY97]. Agents achieving their ideal contracts becomes an optimisation problem of nonlinear functions [KFSBY03a].

Up to date, many negotiation approaches have been proposed in addressing the research problem of issue interdependency in single negotiation. Klein et al. [KFSBY03a] proposed a negotiation approach while considering interdependent issues in single negotiation. Their approach was based on simulated annealing and could achieve near-optimal social welfare [JFL+01, KFSBY03b]. In their approach, the authors defined two types of agents to address the problem of interdependent issues, which were “hill-climbers” and “annealers”. The case of two “annealers” performed better than the case of two “hill-climbers”, and the case of two “annealers” was more likely to get win-win contracts to achieve near-optimal social welfare. Klein et al. gave the first try on addressing interdependent issues in single negotiation. However, each issue in their approach was assigned a binary value, which generated simple binary dependencies between issues. Therefore, the limitation of their approach is that it cannot handle complicated issue interdependency.

Hattori et al. [IKH06, HKI07, IHK07] proposed negotiation protocols for single negotiation with interdependent issues. Their proposed protocols were able to produce near-optimal outcomes for multilateral negotiation. Their approaches outperformed the “hill-climbing” protocol in achieving higher social welfare. However, these approaches required a centralised mediator, so agents could not directly negotiate with each other. Moreover, the scalability of the work while considering a large number of negotiation issues was not well resolved. To address this scalability problem, Fujita et al. [FIK10] proposed a negotiation protocol in achieving high-quality results for multilateral negotiation with interdependent issues. Based on different types of measures for
issue interdependency defined in their work, the interdependencies between issues could be represented by an interdependency graph. By analysing the interdependency graph and grouping issues, Fujita et al.’s protocol outperformed other approaches in reaching higher scalability. However, the issue interdependency in Fujita et al.’s work [FIK10] was considered as binary interdependencies, and centralised control was still required in their work.

The limitations of the approaches above can be summarised as follows. (1) These approaches deal with interdependency by using binary codes between issues in single negotiation, and binary interdependency is not applicable enough in real-world scenarios, and cannot cover all issue interdependency scenarios. (2) Centralised control is required. However, in real-world negotiation scenarios, a decentralised setting should be enabled to let agents negotiate directly with each other.

Alam et al. proposed some approaches [ARR11, ARR13a, ARR13b, AGRR15] on solving the energy exchange problem. In their approaches, the limitations of the binary issue interdependency and the centralised control were resolved. Their proposed negotiation protocol enabled agents to concurrently negotiate over energy exchange with other agents and could lead to Pareto-optimal outcomes. However, their approaches focused on negotiating over multiple issues in single negotiation, and could not be applied to resolve the research problems in MCN as issue interdependency across MCN was not considered in their approaches.

• Issue interdependency across multiple negotiations

Issue interdependency across multiple negotiations indicates that issues in different negotiations are interdependent. This case results in that multiple negotiations are somehow related to each other and issues in different negotiations are involved in the same constraint. So far, some work considered issue interdependency across multiple negotiations, where only single-issue negotiation is involved.

Zhang et al. proposed an approach on multiple related negotiations to solve a negotiation chain problem [ZL07]. Their approach focused on semi-cooperative systems [ZL02, ZL07, ZLA05, ZLP05, ZPL00]. In multiple related negotiations, an agent negotiates with multiple agents on different issues (i.e., one issue is settled through a single-issue negotiation), and the issues in multiple negotiations are interconnected. Each agent has its own goal, and the performances from different agents are related. In negotiation chain scenarios, agents need to perform multiple negotiations to achieve their goals. Zhang
et al. [ZL07] introduced a pre-negotiation approach to allow agents to transfer meta-level information to estimate a negotiation’s success rate. An agent could utilize the information to find a good ordering of all related negotiations. The drawback of their approach is that negotiation time is a key factor to arrange multiple negotiations, which means that the related negotiations have to be performed in a particular order based on the time constraint, so their approach cannot work on MCN, as multiple negotiations involved in MCN are concurrently conducted, and there is no particular order of conducting the negotiations. Zhang et al. made a contribution in solving research problems on multiple related negotiations. Since their approach mainly focused on multiple single-issue negotiations, their approach did not well address the problem of issue interdependency across MCN, where multi-issue negotiation can be involved.

Ren et al. [RZMS10] proposed an approach on a multi-negotiation network. In their work, a multi-negotiation network was proposed to represent the interdependency relationships among multiple negotiations. Their proposed approaches can successfully improve an agent’s overall utility. Even though their work considered the interdependency relationships among different negotiations, their work still focused on the issue interdependency across multiple single-issue negotiations. Moreover, multiple negotiations in their approach were sequentially processed, which could not work on handling the concurrency in MCN.

From the current literature on issue interdependency in agent negotiation, it can be found that most of the current approaches focus on the research of issue interdependency in single negotiation and on issue interdependency between multiple single-issue negotiations. The current approaches cannot work on MCN, and there is very little work on addressing issue interdependency across MCN, where multiple multi-issue negotiations can be involved.

Compared with issue interdependency across multiple single-issue negotiations, the case of issue interdependency across multiple multi-issue negotiations is more complicated. In Figure 2.1, an example of issue interdependency across multiple negotiations is given. Figure 2.1(a) shows the case of issue interdependency across multiple single-issue negotiations, while Figure 2.1(b) shows the case of issue interdependency across multiple multi-issue negotiations. In Figure 2.1, “Ne_i” indicates Negotiation i, “Is_ij” indicates the jth issue in Negotiation i, and an edge between two issues indicates an issue interdependency between the two issues. From Figure 2.1, it can be seen that it is more difficult to handle the issue interdependency across multiple multi-issue negotiations as more constraints exist between issues in
different negotiations and interdependency relationships between different negotiations are more complex. However, this is still a challenge and not well considered in current research.

To overcome the limitation of current work, this thesis proposes a model for MCN in Chapter 3 to handle issue interdependency across MCN. Moreover, issue interdependency across MCN while considering multiple multi-issue negotiations is also considered in Chapter 5 and Chapter 6 of this thesis.

2.2 Concurrent Negotiation

This section reviews related work on concurrent negotiation [NJ04, DH05b, AGL16]. The concurrency in agent negotiation can exist in both single negotiation and multiple negotiations. The concurrency in single negotiation indicates that an agent concurrently negotiates with multiple opponents over one item. The concurrency in multiple negotiations indicates that an agent concurrently negotiates with multiple opponents over multiple items.

- **Concurrency in single negotiation**

  The concurrency in single negotiation happens in multilateral negotiation [WF10], e.g., a buyer agent negotiates with multiple seller agents concurrently. In order to handle such concurrency, Nguyen and Jennings [NJ03a, NJ03b] proposed a model for concurrent negotiation. In detail, there are two components in the model for a buyer agent, which are a coordinator and some negotiation threads (see Figure 2.2). The negotiation threads deal with the exchange of offers with sellers, and the coordinator decides the negotiation strategy for each negotiation thread. During the negotiation, the coordinator

---

**Figure 2.1:** An example of issue interdependency across multiple negotiations
may change the negotiation strategy based on the status of the offers in each negotiation thread.

Most of the current work [NJ04, NJ06, MKV10, WRGJ12] addressed the concurrency in single negotiation by employing the system architecture introduced in Figure 2.2. Napoli et al. [NNR15] presented a variation of the orthogonal bidding strategy in multi-issue negotiation for service composition, and introduced the notion of a weighted reference point to ensure the possibility to find an agreement. By using Napoli et al.'s strategy, the negotiation can be concurrently conducted to avoid a great increase in negotiation time when the number of services involved in the negotiation increases. Nguyen et al. did an extension from their previous work to enhance the coordinator part [NJ04, NJ06]. Specifically, in [NJ04], a buyer agent took its beliefs on the potential sellers into consideration in selecting its initial negotiation strategy. In [NJ06], a commitment manager was added in the buyer agent module, which could reason about commitment and decommitment for the agreements. Compared with their work in [NJ03b, NJ04], the commitment manager was newly introduced to deal with commitment and decommitment [ALS08]. Their new approach could increase the flexibility, enabling seller agents to renege from committed deals with different levels of penalties.

The approaches above mainly dealt with the concurrency in single negotiation, where an agent could concurrently negotiate with multiple opponents
over exactly one item. However, the concurrency in MCN requires that an agent may concurrently negotiate with opponents over multiple items, where the multiple items are interdependent. Due to the interdependency between multiple negotiations, these approaches cannot be directly applied to solving the concurrency in MCN.

- Concurrency in multiple negotiations

In current literature, some work [DH06, SS10, MK14a, MK14b, MK15] has been proposed on addressing the concurrency in multiple negotiations. Shintani et al. [SIS00] proposed an approach to address the concurrency in multi-agent negotiation on a distributed meeting scheduler. In their approach, the authors employed a cloning technique to achieve the concurrency of multiple negotiations. Dang et al. [DH06] proposed a negotiation protocol for concurrent many-to-many negotiations on internet-based services. In many-to-many negotiations, agents interact with different opponents in multiple negotiations, where each negotiation is a bilateral negotiation. By employing Dang et al.’s negotiation protocol, multiple agents can simultaneously negotiate with other agents. Mansour et al. presented a series of approaches [MKW12, MK14a, MK14b, MK15], which focused more on coordinating concurrent negotiations over multiple objects. Mansour et al. [MK14a] focused on a complex negotiation scenario where a buyer negotiated with multiple sellers over different objects through concurrent negotiation. Through applying their proposed approaches, the concurrency in negotiations can be effectively handled.

These above approaches dealt with the general concurrency in multiple negotiations. However, these approaches are not suitable for handling MCN as issue interdependency across MCN are not considered. Let us recall the main features of MCN summarised in Section 1.2 of Chapter 1, which are: (1) the concurrency in MCN, and (2) the issue interdependency across MCN. Achieving the concurrency in MCN is one of the steps of handling MCN, and appropriately handling issue interdependency across MCN must be resolved to completely achieve handling MCN. Issue interdependency across multiple negotiations is a very important factor to impact the negotiation outcome of conducting MCN. In order to overcome this limitation, in this thesis, the analysis of issue interdependency across MCN is paid great attention. A comprehensive negotiation model for MCN is proposed in Chapter 3 to handle the concurrency in MCN. Based on the proposed MCN model, a Colored Petri Net-based protocol is proposed to conduct MCN concurrently. Additionally,
in Chapter 5 and Chapter 6, transition systems are employed in the proposed solutions in this thesis to handle the concurrency in MCN, where multiple multi-issue negotiations are involved.

## 2.3 Dynamic Changing Negotiation Environments

This section reviews related work on dynamic changing negotiation environments in agent negotiation. In this thesis, *a dynamic changing negotiation environment in single negotiation* indicates that the number of agents involved in a negotiation could be changed during the process of the negotiation, while *a dynamic changing negotiation environment in MCN* indicates that the number of negotiations involved in MCN could be changed during the MCN process.

- **Dynamic changing negotiation environments in single negotiation**

  One of the typical scenarios of a dynamic changing negotiation environment in single negotiation is that agents may enter into or leave from a negotiation freely during the process of multilateral negotiation. From current literature, some approaches [LGS06, VAM+10, RZ14] have been proposed to address dynamic changing negotiation environments in single negotiation.

  Moon et al. [MPSK08] proposed a dynamic multi-agent system for task allocation, where their negotiation protocol allows agents to enter into and leave from e-markets during the negotiation. Moon et al.’s approach took the dynamic changes into consideration, but it still focused on single negotiation. An et al. [ALIZ10, ALS11] proposed several approaches for resource allocation, where agents could concurrently conduct negotiations with other agents on multiple resources. In their approach [ALIZ10], the proposed negotiation protocol was extended from the alternating offers protocol [Rub82], where a pair of actions “confirm” and “cancel” was introduced to allow a buyer’s decommitment from an agreement. With these two actions, a buyer was able to choose only one contract when conducting the negotiation with multiple sellers at the same time. By applying their proposed approach, the dynamic changes during the process of the resource allocation, such as multiple opportunities for contracts, the competition of resources and the decommitment, could be effectively handled. In [ALS11], a time-dependent negotiation strategy was applied for agents to respond to the changes of the markets, e.g., the number of requesters on the same resource, the dynamic negotiation situations of other negotiations involving the same resource. Moon et al. and An et al. both considered a dynamic environment where the number of agents could be changed during the process of the negotiation. However, during the negotiation, an
agent may require allocating more resources/tasks, and the newly required resources/tasks are interdependent with the resources/tasks which are under negotiation. In this scenario, the dynamic changes happen, which results in a change of the number of interdependent negotiations during the process of the negotiations. This indicates a dynamic changing negotiation environment in MCN, however, Moon et al.’s work and An et al.’s work cannot handle such a dynamic changing negotiation environment in MCN.

Li et al. [LSG05, LGS06] presented a bilateral negotiation model for a buyer agent while considering the uncertainties and dynamic changes in a negotiation environment. Their model contains three modules, which are a single-threaded module, a synchronized multi-threaded module, and a dynamic multi-threaded module. The single-threaded module governs the negotiation strategy without considering the dynamic changes. The synchronized multi-threaded module deals with the impact from other negotiation threads, which means that a buyer agent makes decisions according to the existing status of the negotiation environment. The dynamic multi-threaded module deals with the dynamic changes coming in the future, which means a buyer agent can react to the existing status of the negotiation environment and reacts to the possible coming dynamic changes through predictions on estimating the expected utility from the negotiation. The single-threaded module deals with bilateral negotiation without considering dynamic changes in the negotiation environment. The synchronized multi-threaded module deals with multilateral negotiation without considering dynamic changes in the negotiation environment. The dynamic multi-threaded module deals with multilateral negotiation by considering dynamic changes in the negotiation environment, where an agent’s opponent could enter into or leave from the environment. Even though their dynamic multi-threaded module could handle dynamic changes of the number of agents in single negotiation, Li et al.’s work could not be applied to handle the dynamic environments in MCN as concurrency, issue interdependency and dynamic changes of the number of negotiations were not considered in their work.

Another real-world scenario of a dynamic environment is cloud computing [AFG10, ZCB10], where Service Level Agreement (SLA) negotiations [ZM11, OGM05] are conducted between customers and service providers. An SLA defines different aspects of web-based services between service providers and consumers. These aspects of the services, i.e., qualities of service properties (e.g., price, responding time, failure possibility, etc.), are important in cloud services [ZZL14]. Specifically, in service-oriented cloud computing, a
customer might ask for a cloud service by processing an SLA negotiation with multiple service providers. During the negotiation, dynamic changes may happen, where the service providers may enter into or leave from the environment. In order to have a solution to such a dynamic environment for an SLA negotiation, an appropriate negotiation approach is necessary. Until now, there are some approaches [AH16, BYST16, OTB+15] focusing on the SLA negotiation.

Dastjerdi et al. [DB15] proposed an approach to focus on the SLA negotiation strategy, which could dynamically adapt to increase profits for cloud providers. Yaqub et al. [YYW+14] proposed a negotiation strategy for agents to efficiently create near-optimal SLAs under time constraints. Copil et al. [CMS+12] proposed an SLA negotiation protocol to obtain a balance between the energy consumed and the performance offered in the cloud. Zan et al. [XCYM11] proposed a policy-based framework to support dynamic SLA negotiations for web services. Their approach focused on bilateral negotiation, where negotiation agents were dynamically created to perform SLA negotiations. Zulkemine et al. [ZM11] presented an SLA negotiation system for web services and proposed a negotiation broker framework to conduct bilateral SLA negotiations based on each party’s requirements. These approaches concentrated on bilateral or multilateral SLA negotiation where only one negotiation goal was involved. However, in a cloud computing environment, an agent may have different goals when applying for multiple cloud services, where each service corresponds to one individual goal. Therefore, these approaches are not applicable for handling MCN in a cloud computing environment.

From current literature, it has reached great achievements in the research of dynamic environments in single negotiation. However, in MCN, a dynamic changing negotiation environment indicates that any negotiation can be added, and any ongoing negotiation can be removed during the process of MCN. These above approaches did not provide solutions to such a dynamic changing negotiation environment in MCN.

- A dynamic changing negotiation environment in MCN

As described previously in Section 1.3, a dynamic changing negotiation environment in MCN indicates that the number of negotiations could be changed during the MCN process. In multiple SLA negotiations, interdependency relationships exist between SLA negotiations, where each service’s process somehow impacts the process of other services. During the negotiations, customers may change their original requests, i.e., adding new service requests or cancelling ongoing service requests. These two features make further difficulties
CHAPTER 2. LITERATURE REVIEW

for handling SLA negotiations in an open and dynamic environment.

From current literature, most of the work focused on dynamic changes in single negotiation, and there is little work considering a dynamic changing negotiation environment in MCN. In order to address this problem, a negotiation protocol is proposed in Chapter 4 to handle a dynamic changing negotiation environment in MCN. In the proposed approach, an updating mechanism is presented to allow negotiations to be added and removed during the process of MCN.

2.4 Negotiation Procedures

This section reviews related work on negotiation procedures for agent negotiation. A negotiation procedure for single negotiation with multiple issues indicates how multiple issues are settled, while a negotiation procedure for multiple negotiations indicates how multiple negotiations are settled.

- Negotiation procedure for single negotiation
  
  Fershtman [Fer00] proposed two procedures for single negotiation, which are: (1) simultaneously and independently negotiating issues, and (2) sequentially negotiating issues, namely, when one issue is settled, another issue will be processed. However, Fershtman did not analyse which procedure is optimal for single negotiation. In order to address this problem, Fatima et al. proposed approaches [FWJ06a, FWJ06b] on the analysis of the optimal procedure among the three well-known negotiation procedures, which were the sequential procedure, the simultaneous procedure, and the package deal procedure. Different negotiation procedures can process multiple issues in single negotiation in different ways.

  - In the sequential procedure, multiple issues are settled one after another. When an agreement on an issue is reached, the outcome on this issue is fixed. Then, other issues are processed one by one. Usually, the sequential procedure is applied for the case of independent issues in single negotiation. The sequential procedure reduces the complexity of the negotiation procedure. However, it may not generate an optimal negotiation outcome.

  - In the simultaneous procedure, multiple issues are processed simultaneously but independently with each other. Usually, the simultaneous procedure is applied for the case of independent issues in single negotiation.
- In the package deal procedure, multiple issues are settled together as a bundle. Agents have to either accept or reject the offer for the whole package. Compared with the previous two procedures, the package deal procedure fits more in the case of interdependent issues in single negotiation. The package deal procedure processes all issues by bundling them as a package, which allows making tradeoffs between all the issues.

Fatima et al. [FWJ06a] did the research on bilateral multi-issue negotiation. Through theoretical analysis in both cases of independent issues and interdependent issues, the authors comprehensively compared the three procedures regarding the time complexity, the Pareto optimality of the equilibrium, the uniqueness of the equilibrium and the time of the agreement. Their work suggested that the package deal procedure would generate Pareto optimal outcome by comparing the outcomes of the three procedures. However, even though the package deal procedure could generate the optimality of the utility achieved by an agent, the computational cost is relatively high with the increase of negotiation issues. Fatima et al. did not address the computational complexity of the package deal procedure when the number of issues is large. In order to solve this issue, Dang et al. [DH05a] proposed an approach on a strategy for multi-issue negotiation. The authors presented a coalition deal negotiation to balance the sequential procedure and the package deal procedure while considering the computational cost and the utility optimality. The approaches above proposed solutions to settle issues in single negotiation only, without considering how to settle multiple negotiations appropriately during the process of MCN. Therefore, these current approaches cannot be applied to MCN in arranging multiple negotiations.

- Negotiation procedure for multiple negotiations

Currently, Some approaches have been done on procedures for single negotiation. However, there is little work on negotiation procedures for multiple negotiations. Negotiation procedures for multiple negotiations are important and necessary for dealing with multiple negotiations in the real world. Let us recall the real-world MCN scenario described in Section 1.1.3. Through using the Internet, a travel agent would like to interact with other agents for a trip package with a budget and time requirement. The trip package may include a flight, accommodation and some tourist activities. Typically, the agent can achieve its goal through concurrently conducting multiple negotiations for flights, accommodations and activities. These negotiations are obviously interdependent with each other while considering factors of the
budget, the sequence and the time. During the negotiations, the agent’s goal is to achieve a series of negotiation agreements with the lowest price and a suitable schedule to satisfy the budget and the time requirement. In order to achieve such a goal, an appropriate negotiation procedure for conducting these multiple negotiations must be used. However, it is still a challenging problem.

In single-issue negotiation, the space of a solution to the negotiation is 1-dimensional. In multi-issue negotiation (e.g., $m$ issues under negotiation), the space of a solution to the negotiation is $m$-dimensional. However, in MCN (e.g., totally $n$ negotiations and at most $m$ issues under negotiation in every single negotiation), the dimensionality of the solution space in MCN could be in the range $[m, m^n]$, where the solution space depends on the complexity of issue interdependency across MCN. It can be found that solving MCN is much more complex than solving single negotiation. Therefore, proposing an appropriate negotiation procedure for MCN, i.e., having a solution to settling all involved issues in MCN, is a very challenging problem.

In order to overcome the limitations of the current work, three feasible negotiation procedures are proposed for MCN in Chapter 5, which are the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. The concurrent negotiation procedure processes multiple negotiations concurrently. The successive negotiation procedure processes multiple negotiations one after another. The clustered negotiation procedure processes multiple negotiations in a more complicated way, where (1) multiple negotiations are portioned into several clusters, (2) negotiations in a cluster are processed by using the concurrent negotiation procedure, and (3) all clusters are processed by using the successive negotiation procedure.

2.5 Reputation-Based Systems

Trust-based systems [KFR08, STW+09] have been paid more and more attention by researches in different research areas. As a type of trust-based systems, a reputation-based system [JIB07, ZH07] helps evaluate an agent’s trustworthiness through the agent’s reputation. Briefly speaking, a reputation-based system is a program, which allows participants involved in the system to rate each other so as to build trust through the participants’ reputations. In centralised systems, the security of a system depends on centralised authenticated control, while in a decentralised system, there is no central control. In order to acquire security in decentralised systems, reputation-based systems can be applied where no central authority is needed. Reputation-based systems have been used in multi-agent systems, especially elec-
tronic commerce, where agents need to make decisions on whether or not to trust contracts.

From current literature, the study of reputation-based systems has been broadly done in many different research fields, such as e-commerce [LRP06], social networks [LS14, HGBY17], wireless sensor networks [AAR+13, ZNLY15], and cloud computing [Man15, YLWV17], etc. In these research fields, the environments are similar, where chances and threats both exist in the environments. A reputation-based system would be helpful for an agent in evaluating its opponent’s trustworthiness and predicting the opponent’s future behaviours, which could help achieve purposes of selecting trustable opponents and avoiding cheating [XL03, RJSG04]. In agent and multi-agent systems, reputation-based systems usually employ personal experience and other agents’ experiences in the decision-making on the trust about an agent.

Ramchurn et al. [RSGJ03] proposed an approach on a computational trust model for interactions between agents based on confidence and reputation. In their approach, confidence and reputation were both considered to assess an agent’s trustworthiness. Confidence depicted an agent’s opinion on the reliability of its opponent, while reputation was derived from the experiences of other agents in the society. When an agent evaluated its opponent’s trust level, both conference and reputation were considered as a combined value. The combined value could be utilised by the agent to decide its negotiation partner. Their work mainly focused on calculating trust level for agents by combining an agent’s confidence and reputation of the agent’s opponent rather than only considering an agent’s subjective view. Their approach could help agents in selecting trustable partners. However, in an agent society, estimating a trust level of an agent through multiple sources of reputations would be more accurate and could lead agents in the society to perform well in order to maintain good reputations, comparing to only considering confidence and reputation. Under this consideration, more and more researchers focused on evaluating an agent’s reputation by considering various information sources from an agent society [HJS06b].

Huynh et al. [HJS06a] proposed a reputation-based trust approach for multi-agent systems. In their approach, a trust and reputation model was presented to comprehensively assess agent’s performance through integrating various information sources in multi-agent systems. Noor et al. [NSY+16] proposed an approach for trust management in cloud computing by employing a reputation-based framework. Their approach relied on a credibility model which could tell the difference between malicious and credible feedbacks to assess the trustworthiness of cloud services. Sabater et al. [SS02] presented an approach on a reputation system for the social network through employing the social dimension of reputation, where the social dimension of reputation means the combined reputation derived from a target agent, neigh-
bours of the target agents and a default system reputation. By considering different sources of reputations, agents in the social network would be led to have a good reputation. Bonatti et al. [BOSM+14] gave an analysis on the integration of reputation with negotiation. The authors mentioned that the integration of reputation with negotiation could contribute to helping select the right opponent as a negotiator. For the purpose of selecting an opponent as a negotiator, employing trust/reputation-based models in agent negotiation to evaluate the opponent is similar to evaluating other features of the opponent. Due to this reason, current trust/reputation-based models would be helpful in selecting trustable opponents for an agent in conducting MCN. However, in an agent society of conducting MCN, different agents have different goals, and an agent’s offer-generation strategy could be affected by issue interdependency across MCN and by opponents’ reputations. Moreover, during the MCN process, the reputation of an agent’s opponents may be changing based on the opponents’ behaviours with other agents. The dynamic changing reputations would probably affect the agent’s offer-generation strategy frequently. Therefore, the current reputation-based approaches are not effective enough and not suitable for handling MCN in an agent society.

In an agent society with conducting MCN, multiple agents interact with each other in order to achieve their own goals through negotiations. For a long-term relationship, agents would like to establish stable relationships with other agents. Under this consideration, the “reputation-based trust” would be suitable to solve this problem in an agent society. Due to the complex environment in conducting MCN in an agent society, reputation-based approaches in an agent society have to be well designed as (1) agents in a society may seek their opponents with long-term relationships and (2) the changing reputations of an agent’s opponents would affect the agent’s offer-generation strategy during the MCN process. In order to address such a challenging issue and overcome the limitations of the current work, a trust-based approach for conducting MCN in an agent society is proposed in Chapter 6, where agents conduct MCN with other agents while considering their opponents’ reputations. The proposed trust-based approach can effectively and efficiently handle social factors of MCN in an agent society.

2.6 Summary

In this chapter, the literature on the five major research issues in MCN was thoroughly reviewed. Particularly, the detailed review focused on issue interdependency, concurrent negotiation, dynamic changing negotiation environments, negotiation procedures and reputation-based systems. Additionally, the advantages and shortcomings of these approaches were analysed at the end of each section. Furthermore,
the advantages of the solutions, which are proposed to solve these research issues in this thesis, were outlined against the limitations in the related approaches.
Chapter 3

A Multiple Concurrent Negotiations Model

To attain Objective 1 of this thesis (refer to Section 1.5), this chapter proposes a model for Multiple Concurrent Negotiations (MCN) to handle the concurrency in MCN and the issue interdependency across MCN, where multiple single-issue negotiations are involved. First, the proposed model formally defines MCN, and the issue interdependency. By employing a graph-based theory, MCN is mathematically represented. In the proposed approach, three algorithms are presented to convert an MCN’s graph-based representation to its Colored Petri Net (CPN)-based representation. By applying manipulations on the CPN-based representation of MCN, a CPN-based negotiation protocol is proposed to conduct MCN.

The outline of this chapter is as follows. Section 3.1 gives problem formulation of MCN, which includes formalised concepts and a classification of MCN. Section 3.2 presents a graph-based representation of MCN. Section 3.3 proposes a CPN-based negotiation protocol for MCN. Section 3.4 demonstrates experimental results by using the proposed MCN model. Based on different negotiation scenarios, Section 3.5 presents some case studies. Section 3.6 summarises this chapter.

3.1 Problem Formulation

In this section, problem formulation of MCN is introduced, which includes some definitions and a classification of MCN.

3.1.1 Definitions

In this subsection, some concepts of MCN are defined, which includes MCN, the issue interdependency, and the multi-negotiation round.
Definition 3.1.1. (Multiple Concurrent Negotiations (MCN)) MCN \( \mathbb{N} \) is defined as a set \( \mathbb{N} = \{ \mathbb{R}_0, \mathbb{R}_1, \cdots \mathbb{R}_i, \cdots \mathbb{R}_n \} (i \geq 0) \), where \( \mathbb{R}_i \) indicates a Multi-Negotiation Round (MNR) and each MNR contains a number of single-issue negotiations (see Definition 3.1.3).

The definition of MCN describes a situation where an agent performs a number of negotiations concurrently, and each negotiation has a negotiation goal, negotiation opponents, and negotiation issues. However, these separated negotiations are somehow related and may impact each other in achieving an overall goal. The issue interdependency between bilateral single-issue negotiations is defined as follows.

Definition 3.1.2. (Issue Interdependency) For single-issue Negotiations \( A_i, A_j \in \mathbb{N} \), \( A_i \propto A_j \) indicates that the issue in Negotiation \( A_j \) depends on the issue in Negotiation \( A_i \), and Negotiation \( A_i \) must start before Negotiation \( A_j \) in each multi-negotiation round (see Definition 3.1.3). The issue interdependency between single-issue negotiations has two properties and one lemma as follows.

Property 1. (Unidirectionality) For \( \forall A_i, A_j \in \mathbb{N} \), if \( A_i \propto A_j \) and \( A_j \propto A_i \), then \( i = j \).

This property indicates that the issue in Negotiation \( A_i \) and the issue in Negotiation \( A_j \) cannot depend on each other simultaneously.

Property 2. (Transitivity) For \( \forall A_i, A_j, A_k \in \mathbb{N} \), it holds that:

\[ A_i \propto A_j, A_j \propto A_k \Rightarrow A_i \propto A_k \]

This property indicates that the issue interdependency between multiple negotiations can be transferred in a single direction.

Lemma 1. For \( \forall A_i, A_j, A_k \in \mathbb{N} \), it holds that: \( A_i \propto A_j, A_j \propto A_k \nRightarrow A_k \propto A_i \).

proof. According to Property 2, it holds that \( A_i \propto A_j, A_j \propto A_k \Rightarrow A_i \propto A_k \) and based on Property 1, if \( A_i \propto A_k \) is held, then \( A_k \propto A_i \) is not held, so Lemma 1 is held.

Based on the issue interdependency between negotiations, the connection between a series of related negotiations can also be defined, and such a connection is called as a multi-negotiation round in this chapter.

Definition 3.1.3. (Multi-Negotiation Round (MNR)) MNR \( \mathbb{R}_i \) is defined as a set \( \mathbb{R}_i = \{ r_{i,0}, r_{i,1}, \cdots r_{i,j}, \cdots r_{i,k_i-1} \} (k_i \geq 1) \), where \( r_{i,j-1} \) indicates the \( j \)th negotiation in MNR \( \mathbb{R}_i \), and \( k_i \) indicates the total number of negotiations involved in MNR \( \mathbb{R}_i \).

In MCN \( \mathbb{N} \), it satisfies \( \bigcup_{i=0}^{l-1} \mathbb{R}_i = \mathbb{N} \), where \( \mathbb{R}_i \) indicates an MNR in MCN \( \mathbb{N} \), and \( l \) is the total number of MNRs. In MNR \( \mathbb{R}_i = \{ r_{i,0}, r_{i,1}, \cdots r_{i,j}, \cdots r_{i,k_i-1} \} \), it
satisfies \( r_{i,0} \propto \cdots \propto r_{i,j} \propto \cdots \propto r_{i,k_i - 1} \).

In an MNR, if all involved negotiations reach a successful negotiation outcome, the MNR is successful. Obviously, the success of MCN is based on the success of all involved MNRs; then, the overall outcome of conducting MCN is called a multi-negotiation outcome in this work, and it is classified as the complete success, the partial success, and the no success based on the expected success on the number of MNRs, where

- MCN achieve the complete success if all MNRs in MCN are successful,
- MCN achieve the partial success if not all but at least one MNR in MCN is successful, and
- MCN achieve the no success if none of MNRs involved in MCN is successful.

In MCN, an agent should have an overall goal before the commencement of conducting the MCN. Based on the goal, an agent makes autonomous decisions to optimally achieve its goal. According to different multi-negotiation outcomes, the agent could have different goals for conducting the MCN. In this chapter, an agent’s overall goal towards conducting MCN is called a Multi-Negotiation Goal (MNG), and it is classified as a complete success goal and a partial success goal.

- A complete success goal indicates that an agent requires an outcome of the complete success in MCN, and
- A partial success goal indicates that an agent requires an outcome of the partial success in MCN.

Let \( \Omega_N \in [0, 1] \) donate the value of an MNG towards conducting MCN, where \( \Omega_N = 1 \) indicates a complete success goal, and \( \Omega_N \in (0, 1) \) indicates a partial success goal.

### 3.1.2 A Classification of MCN

According to the relationships between MNRs, MCN is categorised into the following five types (see Figure 3.1).

1. **Single MCN**: A single MCN \( N \) indicates that MCN include only one MNR and the MNR includes only one negotiation, i.e., \( (R_0 = N) \land (|R_0| = 1) \) (Note: this indicates the presentation of a single negotiation).

2. **Sequential MCN**: A sequential MCN \( N \) indicates that MCN include only one MNR and the MNR includes more than one negotiation, i.e., \( (R_0 = N) \land (|R_0| > 1) \).
3. Synchronized MCN: A synchronized MCN \( N \) indicates that MCN contain multiple MNRs and all MNRs start from the same single negotiation, i.e., for \( \forall R_i \in \mathbb{N}, l > 1, r_{0,0} = \cdots = r_{l,0} = \cdots = r_{l-1,0} = \bigcap_{i=0}^{l-1} R_i \).

4. Merging MCN: A merging MCN \( N \) indicates that MCN contain multiple MNRs and all MNRs end with the same single negotiation, i.e., for \( \forall R_i \in \mathbb{N}, l > 1, r_{0,k_0-1} = \cdots = r_{i,k_i-1} = \cdots = r_{l-1,k_{l-1}-1} = \bigcap_{i=0}^{l-1} R_i \).

5. Hybrid MCN: A hybrid MCN \( N \) indicates a complex MCN, which is a combination of a sequential MCN, a synchronized MCN and a merging MCN, i.e., for \( \forall R_i \in \mathbb{N}, l > 1, \bigcap_{i=0}^{l-1} R_i \neq \emptyset \).

Based on the descriptions above, it can be found that a synchronized MCN or a merging MCN is a special case of a hybrid MCN, where it either starts from or ends with the same negotiation.

3.2 A Graph-Based Modelling for MCN

In order to model MCN, an MCN’s representation based on a graph-based theory [BM76] is proposed in this section, which contains a graph-based representation, a logic-based representation and a utility calculation for MCN.

3.2.1 A Graph-Based Representation of MCN

Definition 3.2.1. (Graph-Based Representation) MCN \( N \) is represented by a directed graph \( G = (V, E) \), where \( V = \mathbb{N} \) indicates vertexes in \( G \), and \( E \) indicates the issue interdependency between negotiations in MCN \( N \). For any two Negotiations \( A_i \) and \( A_j \), if \( A_i \preceq A_j \), then \( e_{ij} = (A_i, A_j) \in E \).

Based on the classification of MCN, the graph-based representations of five types of MCN are shown in Figure 3.1.

3.2.2 A Logic-Based Representation of MCN

In order to indicate the logical relationships between a single negotiation’s outcome and the overall negotiation outcome of MCN, logic-based representations of Negotiation \( A_i \), MNR \( R_i \) and MCN \( N \) are defined as follows.

Definition 3.2.2. (Logic-Based Representation) Let \( L(A_i) = \{True, False\}(i \geq 0) \) be a logic-based representation of Negotiation \( A_i \), where \( L(A_i) = True \) indicates that Negotiation \( A_i \) is successful, \( L(A_i) = False \) indicates that Negotiation \( A_i \) fails.
Definition 3.2.3. (Logical Conjunction) Let $L(R_i)$ be a logic-based representation of MNR $R_i$, then $L(R_i) = \bigcap_{A_j \in R_i} L(A_j)$ ($i \geq 0, j \geq 0$). $L(A_i) \cap L(A_j)$ indicates the logical conjunction relationship of $A_i \propto A_j$. $L(A_i) \cap L(A_j)$ has the same truth value table as $L(A_i) \wedge L(A_j)$ (logic “and”), but the symbol “$\cap$” does not satisfy associativity and commutativity due to the properties of unidirectionality and transitivity in Definition 3.1.2.

Definition 3.2.4. (Logical Disjunction) Let Equation $L(N) = \bigvee_{\forall R_i \in N} L(R_i)$ ($i \geq 0$) indicate a logic-based representation of MCN $N$, where “$\bigvee$” indicates a logical disjunction of MNR $R_i$ in MCN $N$. The symbol “$\bigvee$” and logical symbol “$\lor$” (logic “or”) have the same truth value table, but the symbol “$\bigvee$” does not satisfy associativity and commutativity.

Based on the logic-based representations introduced above, the success possibility of MCN $N$ can be further estimated as follows.
Let $V(A_i)$ indicate the success possibility of Negotiation $A_i$, and $V(A_i)$ is calculated by Equation (3.1) based on Negotiation $A_i$’s logic-based representation.

$$V(A_i) = \begin{cases} 
0 & \text{if } L(A_i) = False, \\
1 & \text{if } L(A_i) = True. 
\end{cases} \quad (3.1)$$

There are two cases, where $V(A_i) = 0$ means that Negotiation $A_i$ fails and $V(A_i) = 1$ means that Negotiation $A_i$ succeeds.

Based on Equation (3.1), the success possibility of MNR $R_i$ and MCN $N$ are calculated by Equation (3.2) and Equation (3.3), respectively.

$$V(R_i) = \prod_{\forall A_j \in R_i} V(A_j) \ (i \geq 0, j \geq 0) \quad (3.2)$$

$$V(N) = \sum_{\forall R_i \in N} \frac{V(R_i)}{l} \ (i \geq 0) \quad (3.3)$$

where $l$ is the total number of MNRs in MCN $N$.

### 3.2.3 Utility Calculation for MCN

In this subsection, a utility calculation method for MCN is presented based on the utility achieved from MNRs as follows.

Let $U(A_i)$ be the utility achieved from Negotiation $A_i$. Based on the definition of $V(A_i)$, the utility achieved from MNR $R_i$ is calculated by Equation (3.4) as follows.

$$U(R_i) = \sum_{\forall A_j \in R_i} \left( \omega_j \times U(A_j) \times V(A_j) \right) \ (i \geq 0, j \geq 0), \quad (3.4)$$

where $U(R_i) \in [0, 1]$, $\omega_j \in [0, 1]$ represents an agent’s preference for Negotiation $A_j$, $\omega = (\omega_0, \omega_1, \cdots, \omega_j, \cdots, \omega_n)$, and $\sum_{i=0}^{n} \omega_j = 1$.

Based on the logic-based representation of $V(R_i)$ and the utility achieved from MNR $R_i$, the utility achieved from MCN $N$ is calculated by Equation (3.5) as follows.

$$U(N) = \sum_{\forall R_i \in N} \left( U(R_i) \times V(R_i) \right) \quad (3.5)$$

where $U(N) \in [0, 1]$, $i \geq 0$ and $l$ is the total number of MNRs in MCN $N$.

### 3.3 A CPN-Based Negotiation Protocol for MCN

In this section, a CPN-based negotiation protocol is proposed to concurrently conduct MCN by considering issue interdependency.
3.3.1 The Connection between CPN and MCN

The Colored Petri Net (CPN) is a language for the modelling and validation of systems in which concurrency, communication, and synchronisation play a major role [JKW07, JR12]. A CPN model of a system is an executable model representing the states of the system and the events (transitions) that can cause the system to change states [JK09, Jen13]. Therefore, a CPN is a very useful tool to support concurrent behaviours of systems.

By taking advantages of CPNs, agents can flexibly control the concurrency in MCN. In this thesis, a CPN is defined as a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ [JK09], where $P$ is a set of places, $T$ indicates a set of transitions, $A$ is a set of arcs and $A \subseteq P \times T \cup T \times P$, $\Sigma$ is a finite non-empty colour set, $V$ indicates a set of variables, $C$ indicates a set of colour functions, which assigns colour sets to places, and $C : P \rightarrow \Sigma$ and $C(p) \in \Sigma$, $G$ is a set of guard functions, which assigns guards to transitions, and $E$ and $I$ are a set of arc expression functions and a set of initialisation functions, respectively. Places in a CPN may contain discrete numbers of marks, which are called tokens [JK09]. A transition of a CPN may be fired if it is enabled, i.e., each transition’s input place has at least one token. When a transition fires, it consumes the required input tokens from its input places, and generates tokens to its output places.

In the proposed CPN-based MCN protocol, transitions represent negotiations, and places represent states of negotiations. The inputs and outputs of negotiations are shown by arc directions. Token $1'(A, m)(m \geq 1)$ indicates that one offer is received by an agent in the $m$th negotiation round in Negotiation $A$ (i.e. enabling Transition $t_A$). In MCN, the first place and the last place of each MNR are the initial place and the final place, respectively. Each initial place contains at least one token to fire following transitions. For any MNR $R_i = \{r_{i,0}, r_{i,1}, \ldots, r_{i,j}, \ldots, r_{i,k_i-1}\} (0 \leq j \leq k_i - 1, k_i \geq 2)$ in MCN, a token in the initial place of MNR $R_i$ (i.e., an input place of Transition $t_{r_{i,j}}$) is used to activate MNR $R_i$. If Transition $t_{r_{i,1}}$ is activated by a token, it means that Negotiation $r_{i,1}$ finishes a single negotiation round. If Negotiation $r_{i,j}$ fails, Transition $t_{r_{i,j}}$ will not be activated, and the execution of MNR $R_i$ will be terminated. If Negotiation $r_{i,j}$ succeeds, the token will be used to fire the following transitions in MNR $R_i$ based on arc directions. If all transitions in MNR $R_i$ have been activated, it is called that MNR $R_i$ finishes an MNR negotiation round. In order to handle the concurrency in MCN, a backward arc is added to show that different negotiations are conducted concurrently.

Based on the categories of MCN described in Section 3.1.2, detailed CPN-based
representations of five types of MCN are shown as follows.

1. Single MCN

The CPN-based representation of a single MCN \( N = \{ A_i \} \) is shown in Figure 3.2.

![Figure 3.2](image)

**Figure 3.2:** The CPN-based representation of a single MCN

In Figure 3.2, Transition \( t_{A_i} \) represents Negotiation \( A_i \), and Places \( P_0 \) and \( P_1 \) are an input place and an output place of Transition \( t_{A_i} \), respectively. An initial Token \( 1'(A,1) \) in Place \( P_0 \) acts as an input to activate Transition \( t_{A_i} \). If Transition \( t_{A_i} \) is enabled in the \( m \)th round, it outputs Tokens \( 1'(A,m) \) and \( 1'(A,m+1) \) to Places \( P_1 \) and \( P_0 \), respectively. Token \( 1'(A,m) \) in Place \( P_1 \) acts as a finished mark of the single MCN \( N \). It means that Negotiation \( A_i \) finishes a negotiation round if a token exists in Place \( P_1 \). Token \( 1'(A,m+1) \) makes Transition \( t_{A_i} \) enabled for the next negotiation round of Negotiation \( A_i \). When Negotiation \( A_i \) finishes in the end, the number of tokens in Place \( P_1 \) must be equal to the number of negotiation rounds conducted by Negotiation \( A_i \).

2. Sequential MCN

A sequential MCN \( N \) has only one MNR, and the CPN-based representation of a sequential MCN \( N = \{ A_0, A_1, \cdots A_{k-1} \} (k \geq 2) \) is shown in Figure 3.3.

![Figure 3.3](image)

**Figure 3.3:** The CPN-based representation of a sequential MCN
At the beginning of conducting the sequential MCN \( N \), an initial Token \( 1'(R, 1) \) in the initial place (i.e., Place \( P_0 \)) acts as an input to activate Transition \( t_{A_0} \). If Transition \( t_{A_0} \) is enabled in the \( m \)th round, it outputs Tokens \( 1'(R, m) \) and \( 1'(R, m + 1) \) to Places \( P_1 \) and \( P_0 \), respectively. Token \( 1'(R, m) \) in Place \( P_1 \) is used to activate Transition \( t_{A_1} \), and Token \( 1'(R, m + 1) \) makes Transition \( t_{A_0} \) enabled for the next MNR negotiation round. The activation procedures of posterior transitions are same as Transition \( t_{A_0} \)’s. Once the sequential MCN \( N \) finishes the \( m \)th MNR negotiation round, \( m \) tokens will appear in the final place (i.e., Place \( P_k \)). When the sequential MCN \( N \) finishes in the end, the number of tokens in Place \( P_k \) must be equal to the number of MNR negotiation rounds conducted by the sequential MCN \( N \).

3. Synchronized MCN

A synchronized MCN \( N = \{A_0, A_1, \cdots, A_i, \cdots, A_k\} \) with multiple MNRs \( R_j \), where \( R_j = \{A_0, A_{j+1}\} \) for \( 0 \leq j \leq k - 1 \). The CPN-based representation of a synchronized MCN \( N \) is shown in Figure 3.4.

![Figure 3.4: The CPN-based representation of a synchronized MCN](image-url)
final place (i.e., Place $P_{j+2}$), the synchronized MCN $N$ finishes an MNR negotiation round. When the synchronized MCN $N$ finishes in the end, the number of tokens in Place $P_{j+2}$ must be equal to the number of negotiation rounds of MNR $R_j$. The processes of other MNRs are same as the process of MNR $R_j$, and all MNRs are conducted concurrently.

4. Merging MCN

A merging MCN $N = \{A_0, A_1, \cdots, A_i, \cdots, A_k\}$ ($0 \leq i \leq k, k \geq 2$) has multiple MNRs $R_j$, where $R_j = \{A_j, A_k\}$ ($0 \leq j \leq k-1$). The CPN-based representation of a merging MCN $N$ is shown in Figure 3.5.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{merging_mcns.png}
\caption{The CPN-based representation of a merging MCN}
\end{figure}

For any MNR $R_i = \{A_j, A_k\}$ in the merging MCN $N$, at the beginning, an initial Token $1'(R_j, 1)$ in Place $P_j$ acts as an input to activate Transition $t_{A_j}$. If Transition $t_{A_j}$ is enabled in the $m$th round, Transition $t_{A_j}$ outputs Tokens $1'(R_j, m)$ and $1'(R_j, m+1)$ to Places $P_{j+k}$ and $P_j$, respectively. Token $1'(R_j, m)$ in Place $P_{j+k}$ is used to activate the posterior Transition $t_{A_k}$, and Token $1'(R_j, m+1)$ acts as the input of Transition $t_{A_j}$ for the next MNR negotiation round. If Transition $t_{A_k}$ is
enabled, it outputs Token $1'(R_j, m)$ to Place $P_{2k}$. If a token appears in the final place (i.e., Place $P_{2k}$), the merging MCN $N$ finishes an MNR negotiation round. When the merging MCN $N$ finishes in the end, the number of Tokens $1'(R_j, m)$ in Place $P_{2k}$ must be equal to the number of negotiation rounds conducted by MNR $R_j$. The processes of other MNRs are same as the process of MNR $R_j$, and all MNRs are conducted concurrently.

5. Hybrid MCN

A hybrid MCN is a combination of a sequential MCN, a synchronized MCN and a merging MCN. A hybrid MCN can be used to represent any complex structure of MCN. The CPN-based representation of a hybrid MCN is shown in Figure 3.6.

![Figure 3.6: The CPN-based representation of a hybrid MCN](image)

In Figure 3.6, the CPN-based representation of a hybrid MCN $N$ is illustrated, i.e., $N = \{A_0, A_1, A_2, A_3, A_4\}$ where $R_0 = \{A_0, A_2, A_3\}$, $R_1 = \{A_0, A_2, A_4\}$, $R_2 = \{A_1, A_2, A_3\}$, $R_3 = \{A_1, A_2, A_4\}$. In the CPN-based representation of the hybrid MCN $N$, each initial place (i.e., Places $P_0$ and $P_1$) has two initial tokens, where Tokens $1'(R_{00}, 1)$ and $1'(R_{01}, 1)$ are in Place $P_0$, and Tokens $1'(R_{10}, 1)$ and $1'(R_{11}, 1)$ are in Place $P_1$. The four initial tokens act as different inputs to activate four MNRs (i.e., $R_0, R_1, R_2, R_3$) separately. The process of handling each MNR is same as the process of handling a sequential MCN. In the hybrid MCN $N$, all MNRs are conducted concurrently.

3.3.2 The Conversion of MCN from a Graph-Based Representation to a CPN-Based Representation

In this subsection, algorithms for converting a graph-based representation of MCN to its corresponding CPN-based representation are presented.
Algorithm 1 converts the graph-based representation of a single negotiation to its CPN-based representation.

**Algorithm 1** The Conversion from Negotiation $A_i$’s graph-based representation to its CPN-based representation ($Conv_1(A_i)$)

**Input:** the graph-based representation of Negotiation $A_i$

**Output:** Negotiation $A_i$’s corresponding CPN-based representation $C_{A_i} = (P, T, A, \Sigma, V, C, G, E, I)$

1. $P \leftarrow \{P_0, P_1\}$;
2. $T \leftarrow \{t_{A_i}\}$;
3. $A \leftarrow \{(P_0, t_{A_i}), (t_{A_i}, P_1), (t_{A_i}, P_0)\}$;
4. $\Sigma \leftarrow \{1'(A, 1), 1'(A, 2), \ldots, 1'(A, m), \ldots, 1'(A, n)\}$;
5. $V \leftarrow \{n, A_i\}$;
6. $C(P_0) \leftarrow \varnothing$;
7. $C(P_1) \leftarrow \varnothing$;
8. for $j \leftarrow 1$ to $m$ do
   9. \hskip1em $C(P_0) \leftarrow C(P_0) \cup 1'(A, j)$;
10. \hskip1em $C(P_1) \leftarrow C(P_0) \cup 1'(A, j)$;
11. end for;
12. $C(P_0) \leftarrow C(P_0) \cup 1'(A, m + 1)$;
13. $C \leftarrow \{C(P_0), C(P_1)\}$;
14. $E \leftarrow \{A, m\}$;
15. $I(P_0) \leftarrow 1'(A, m)$;
16. $I(P_1) \leftarrow \varnothing$;
17. $I \leftarrow \{I(P_0), I(P_1)\}$;
18. if Negotiation $A_i$ finishes a negotiation round then
19. \hskip1em $G(t_{A_i}) \leftarrow true$;
20. else if Negotiation $A_i$ does not finish a negotiation round then
21. \hskip1em $G(t_{A_i}) \leftarrow false$;
22. end if;
23. $G \leftarrow G(t_{A_i})$;
24. $M_0(P_0) \leftarrow 1'(A, m)$;
25. $M_0(P_1) \leftarrow \varnothing$;
26. $M_0(P) \leftarrow \{M_0(P_0), M_0(P_1)\}$;
27. return $C_{A_i}$.

In Algorithm 1, the input is the graph-based representation of Negotiation $A_i$, and the output is Negotiation $A_i$’s CPN-based representation $C_{A_i}$. First, Algorithm 1 creates a set of places $P$, a set of transitions $T$, a set of arcs $A$, a colour set $\Sigma$ and a set of variables $V$, where variable $n$ indicates the number of total negotiation rounds (Lines 1-5). The sets of colour functions of Places $P_1$ and $P_2$ are initialised as empty sets (Lines 6-7), and the algorithm gets the real values of the set of colour functions $C$ from the graph-based representation (Lines 8-13). The set of arc expression functions $E$ and the set of initialisation functions $I$ are set up (Lines 14-17). Afterwards, a set of guard functions $G$ and the markings of Places $P_1$ and $P_2$ are set up (Lines 18-26). In the end, Algorithm 1 returns the CPN-based representation $C_{A_i}$ of Negotiation.
Algorithm 2 The Conversion from MNR $R_i$'s graph-based representation to its CPN-based representation $(Conv_2(R_i))$

**Input:** the graph-based representation of MNR $R_i = \{r_{i,0}, r_{i,1}, \ldots, r_{i,j}, \ldots, r_{i,k_i-1}\}$

**Output:** MNR $R_i$'s corresponding CPN-based representation $C_{R_i} = (P, T, A, \Sigma, V, C, G, E, I)$

1: for $j \leftarrow 0$ to $k_i - 1$ do
2: \hspace{1em} $(P^j, T^j, A^j, \Sigma^j, V^j, C^j, G^j, E^j, I^j) \leftarrow Conv_1(r_{i,j});$
3: end for;
4: $T \leftarrow \bigcup_{j=1}^{k_i} T^j;$
5: $P \leftarrow \bigcup_{j=1}^{k_i} P^j;$
6: $P \leftarrow P \cup P_{k_i+1};$
7: $A \leftarrow \bigcup_{j=1}^{k_i} A^j;$
8: for $j \leftarrow 1$ to $k_i - 1$ do
9: \hspace{1em} $A \leftarrow A - \{(t_{r_{i,j}}, P_j)\};$
10: end for;
11: for $j \leftarrow 1$ to $k_i - 1$ do
12: \hspace{1em} $\Sigma \leftarrow \Sigma^0 \cup \Sigma^j;$
13: \hspace{1em} $V \leftarrow V^0 \cup V^j;$
14: \hspace{1em} $C \leftarrow C^0 \cup C^j;$
15: \hspace{1em} $G \leftarrow G^0 \cup G^j;$
16: \hspace{1em} $E \leftarrow E^0 \cup E^j;$
17: \hspace{1em} $I(P_j) \leftarrow \{\varnothing\};$
18: \hspace{1em} $j + +;$
19: end for;
20: $I \leftarrow \{I(P_0), I(P_1), \ldots, I(P_j), \ldots, I(P_{k_i-1})\};$
21: $M_0(P_0) \leftarrow 1'(R_0, m);$  
22: for $j \leftarrow 0$ to $k_i - 1$ do
23: \hspace{1em} $M_0(P_{j+1}) \leftarrow \varnothing;$
24: end for;
25: return $C_{R_i}$.

The input of Algorithm 2 is the graph-based representation of MNR $R_i$, and the output is MNR $R_i$'s corresponding CPN-based representation $C_{R_i}$. At first, Algorithm 1 is employed to convert the graph-based representation of every single negotiation in MNR $R_i$ to its corresponding CPN-based representation (Lines 1-3), and Algorithm 2 gets the set of transitions $T$ and the set of places $P$ (Lines 4-6). In Lines 7-20, Algorithm 2 creates a set of arcs $A$, a colour set $\Sigma$, a set of variables $V$, a set of colour functions $C$, a set of guard functions $G$, a set of arc expression functions $E$ and a set of initialisation functions $I$, respectively. In Lines 21-24, Algorithm 2 generates markings for each place. In the end, Algorithm 2 returns the CPN-based representation $C_{R_i}$ of MNR $R_i$. 

A$_i$ as the output.

Algorithm 2 converts the graph-based representation of MNR $R_i$ to its CPN-based representation.
Algorithm 3 The Conversion from MCN $N$’s graph-based representation to its
CPN-based representation ($\text{Conv}_3(N)$)

**Input:** the graph-based representation of MCN $N = \{A_0, A_1, \cdots A_i, \cdots A_{n-1}\}$

**Output:** MCN $N$’s corresponding CPN $C_N = (P, T, A, \Sigma, V, C, G, E, I)$ based on
the inputs of MCN $N$

1: for $i \leftarrow 0$ to $l - 1$ do
2: \quad $(P^i, T^i, A^i, \Sigma^i, V^i, C^i, G^i, E^i, I^i) \leftarrow \text{Conv}_2(R_i)$;\footnote{In Algorithm 3, a function $f(r_{i,j}) = k$ is defined for searching arcs, where $r_{i,j} = A_k$.}
3: end for;
4: $T \leftarrow \bigcup_{i=0}^{n-1} T^i$;
5: $P \leftarrow \bigcup_{i=0}^{n-1} P^i$;
6: $P \leftarrow P \cup P_{\text{final}}$;
7: $V \leftarrow \{n, A_1\}$;
8: for $k \leftarrow 0$ to $k_i - 1$ do
9: \quad for $k' \leftarrow 0$ to $k_i$ do
10: \quad \quad if $r_{i+1,k'} = r_{i,k}$ then
11: \quad \quad \quad $A \leftarrow A \cup \{(t_{r_{i+1,k'}}, P_{f(r_{i,k})}^{(r_{i,k})+1})\}^a$;
12: \quad \quad \quad $A \leftarrow A - \{(t_{r_{i+1,k'}}, P_{f(r_{i,k})+1})\}$;
13: \quad \quad else $k' + +$;
14: \quad end if;
15: \quad end for;
16: \quad $k + +$;
17: end for;
18: for $j \leftarrow 0$ to $n$ do
19: \quad $C(P_i) \leftarrow C(P_i) \cup 1'(A_i, j)$;
20: end for;
21: $C \leftarrow \{C(P_1), C(P_2), \cdots C(P_{\text{final}})\}$;
22: $I(P_0) \leftarrow 1'(R_0, m)$;
23: for $i \leftarrow 0$ to $l - 1$ do
24: \quad $\Sigma \leftarrow \{1'(A_i, 1), 1'(A_i, 2), \cdots 1'(A_i, m), \cdots 1'(A_i, n)\}$;
25: \quad $G(t_{A_i}) \leftarrow \{\text{true, false}\}$;
26: \quad $G \leftarrow G(t_{A_i})$;
27: \quad $E \leftarrow (A_i, m)$;
28: \quad $I(P_{i+1}) \leftarrow \emptyset$;
29: end for;
30: $I \leftarrow \{I(P_0), I(P_1) \cdots I(P_{\text{final}})\}$;
31: return $C_N$. 
Algorithm 3 converts the graph-based representation of MCN \( N \) to its CPN-based representation. The input of Algorithm 3 is the graph-based representation of MCN \( N \). The output is the corresponding CPN-based representation \( C_N \). Algorithm 2 is first employed to convert every single MNR’s graph-based representation to its corresponding CPN-based representation (Lines 1-3). Algorithm 3 creates a set of transitions \( T \), a set of places \( P \) and a set of variables \( V \) (Lines 4-7). Algorithm 3 generates a set of arcs \( A \) by searching and combining the same transitions in different MNRs (Lines 8-17), and also generates a set of colour functions \( C \) (Lines 18-21). Algorithm 3 generates a set of initialisation functions \( I \), a colour set \( \Sigma \), a set of guard functions \( G \), a set of arc expression functions \( E \) (Lines 22-30). In the end, Algorithm 3 returns MCN \( N \)’s CPN-based representation \( C_N \).

### 3.3.3 A CPN-Based Protocol for MCN

In this subsection, a CPN-based protocol for conducting MCN is proposed. At first, Algorithm 4 presents a protocol for conducting MNR \( R_i \). Then, based on the outcomes of conducting MNR \( R_i \), Algorithm 5 presents a protocol for conducting MCN \( N \). Finally, by comparing values of the utility achieved from MCN \( N \) (i.e., \( U(N) \)) and MNG \( \Omega_N \) (refer to Section 3.1.1), Algorithm 5 generates a final result of conducting MCN \( N \), i.e., success or failure.

Algorithm 4 presents a protocol for conducting MNR \( R_i \).

**Algorithm 4**: A CPN-based protocol for conducting MNR \( R_i \)

**Input**: MNR \( R_i \)’s CPN-based representation, i.e., \( C_{R_i} = (P, T, A, \Sigma, V, C, G, E, I) \).

**Output**: \( V(R_i) \) and \( U(R_i) \).

1. for \( k \leftarrow 0 \) to \( k_i - 1 \) do
2.   if \( G(t_{r_i,k}) \leftarrow false \) then
3.     terminate MNR \( R_i \);
4.     set \( V(r_{i,k}) \leftarrow 0 \);
5.   else if \( G(t_{r_i,k}) \leftarrow true \) then
6.     set \( V(r_{i,k}) \leftarrow 1 \);
7.   end if;
8. end for;
9. calculate \( V(R_i) \) and \( U(R_i) \);
10. return \( V(R_i) \) and \( U(R_i) \).

The input of Algorithm 4 is MNR \( R_i \)’s CPN-based representation, i.e., \( C_{R_i} \), and the outputs of Algorithm 4 are the success possibility of conducting MNR \( R_i \) and the utility achieved from MNR \( R_i \), i.e., \( V(R_i) \) and \( U(R_i) \). Algorithm 4 sets a loop to make a decision on whether a transition is enabled (Line 1). For each Transition
Algorithm 5 proposes a protocol for conducting MCN, where the inputs are MCN's CPN-based representation (i.e., \( C_N \)) and MNG \( \Omega_N \), and the output is the final result of conducting MCN, i.e., success or failure.

**Algorithm 5 A CPN-Based protocol for conducting MCN**

**Input:** MCN N’s CPN-based representation, i.e., \( C_N = (P, T, A, \Sigma, V, C, G, E, I) \), and MNG \( \Omega_N \).

**Output:** success or failure.

1: execute each MNR concurrently by employing Algorithm 4;
2: while at least one MNR \( R_i \) is finished do
3: calculate \( V(N) \) and \( U(N) \);
4: if \( V(N) < \Omega_N \) then
5: keep executing other unfinished MNRs;
6: else if \( V(N) \geq \Omega_N \) then
7: terminate MCN N and quit;
8: return success;
9: end if;
10: end while;
11: while the CPN is not completed do
12: calculate \( V(N) \) and \( U(N) \);
13: if \( V(N) \geq \Omega_N \) then
14: return success;
15: else if \( V(N) < \Omega_N \) then
16: return failure;
17: end if;
18: end while.
19: return success or failure.

In Algorithm 5, Algorithm 4 is first employed to concurrently execute each MNR (Line 1). The values of \( V(N) \) and \( U(N) \) are calculated as soon as executing one MNR \( R_i \) is completed (Lines 2-3). According to the value of \( V(N) \), the algorithm will make a decision. If the value of \( V(N) \) is less than the value of MNG \( \Omega_N \), the algorithm continues the conduct of other unfinished MNRs (Lines 4-5). If the value of \( V(N) \) is equal to or greater than the value of MNG \( \Omega_N \), the algorithm stops conducting MCN N (Lines 6-7). Algorithm 5 shows if the intermediate value of \( V(N) \) is equal to or is greater than the value of MNG \( \Omega_N \), the algorithm outputs “success” as there is
no need to process other unfinished MNRs (Lines 11-14). Otherwise, the algorithm returns “failure” if the final value of $V(N)$ is less than the value of MNG $\Omega_N$ (Lines 15-16). By doing such a trade-off, the efficiency of handling MCN $N$ is significantly improved. In the end, Algorithm 5 outputs the final result of conducting MCN $N$, i.e., success or failure (Line 19).

### 3.4 Experiment

In this section, an experiment is conducted to evaluate the effectiveness and efficiency of the proposed approach for MCN in handling concurrency and issue interdependency.

#### 3.4.1 Experimental Settings

Suppose that Agent $a$ conducts MCN with different opponents. In the experiment, a single-issue negotiation model [FWJ02] is employed to process every single negotiation. Every agent involved in the MCN randomly chooses a concession strategy including Conceder, Linear or Boulware Strategy [FSJ98]. The utility function $U^a$ for every single negotiation is described by Equation (3.6) as follows.

$$U^a(\text{Counter Offer}) = \frac{\text{Reserved Offer} - \text{Counter Offer}}{\text{Reserved Offer} - \text{Initial Offer}}$$  \hspace{1cm} (3.6)

In the experiment, Agent $a$’s preferences for different negotiations are generated randomly. For simplification, every single negotiation is a bilateral negotiation, i.e., Agent $a$ has one opponent in each negotiation. The deadlines for different agents are selected randomly between 10 and 20 rounds.

Figure 3.7 shows MCN $N$’s graph-based representation, where $N = \{A_0, A_1, A_2, A_3, A_4, A_5\}$, and $R_0 = \{A_0, A_1, A_3, A_5\}$, $R_1 = \{A_0, A_1, A_4\}$, and $R_2 = \{A_0, A_2\}$. In the experiment, the MCN in Figure 3.7 is taken to test the performance of the proposed solution in this work.
The parameters for every single negotiation contain its preference, Agent a’s and its opponents’ deadlines, initial offers and reserved offers. The experiment was conducted 100 times, where all parameters were generated randomly each time.

### 3.4.2 Experimental Results

The purpose of the experiment is to test the performance of the proposed approach in terms of effectiveness and efficiency. The experimental results are shown from Figure 3.8 to Figure 3.12.

![Image of Figure 3.7: MCN N’s graph-based representation]

**Figure 3.7:** MCN N’s graph-based representation

In Figure 3.8, the horizontal axis indicates three different outcomes of conducting MCN, and the vertical axis indicates the percentage of each multi-negotiation outcome.
outcome in the experiment. The figure shows the percentages of the Complete Success, the Partial Success and the No Success are 51%, 40% and 9%, respectively. The experimental result indicates that the proposed approach has a high probability to avoid the failure of conducting MCN based on the experimental settings.

Besides the distribution of the multi-negotiation outcomes, other three types of experimental results are also recorded, which are the final utility, the round interval and the number of rounds. The final utility achieved from MCN $\mathbb{N}$ is $U(\mathbb{N})$, which is the overall utility achieved by the agent when MCN $\mathbb{N}$ finishes. The round interval and the number of rounds of MCN $\mathbb{N}$ indicate the number of negotiation rounds spent on a particular negotiation with and without considering other ongoing negotiations in MCN $\mathbb{N}$, and are defined by Equations (3.7) and (3.8), respectively.

Let $T = \{t_0, t_1, \ldots, t_i, \ldots, t_k\}$ indicate MCN $\mathbb{N}$’s round sequence, which means that MCN $\mathbb{N}$ finishes its negotiation round at Rounds $t_0, t_1, \ldots, t_i, \ldots, t_k$ (i.e., every single negotiation finishes its $j$th negotiation round at Round $j - 1$). The round interval of MCN $\mathbb{N}$ is defined as follows:

$$f(T) = t_k - t_0.$$  \hspace{1cm} (3.7)

The number of rounds of MCN $\mathbb{N}$ is defined as follows:

$$g(T) = |T|.$$  \hspace{1cm} (3.8)

To get a good performance of the proposed approach, the value of the final utility should be as high as possible, and the values of the round interval and the number of rounds should be as low as possible.
Figure 3.9 and Figure 3.10 show final utilities, where Figure 3.9 indicates the final utility achieved from every single negotiation and Figure 3.10 illustrates the final utilities achieved from the MNRs and the MCN.

In Figure 3.9, the horizontal axis indicates single negotiations, and the vertical axis indicates final utilities. From Figure 3.9, it can be seen that the minimum values of final utilities are 0, which means that these negotiations fail, while the maximum values of final utilities are about 0.6 (the real maximum values of final utilities for single negotiations depend on agents’ negotiation strategies and deadlines). The values of average final utilities are around 0.4, which shows that the proposed approach performed reasonably and stably in the experiment.
In Figure 3.10, the horizontal axis indicates the MNRs and the MCN, while the vertical axis indicates final utilities. The minimum values of final utilities achieved from the MNRs and the MCN are 0, while the maximum values of final utilities achieved from MNRs vary from 0.32 to 0.44, and the maximum value of the final utility achieved from the MCN is about 0.3. When comparing with the values of final utilities achieved from single negotiations, the values of final utilities achieved from MNRs and the MCN have a bigger range of variation. The main reason is that the preference for every single negotiation is considered in calculating utilities achieved from MNRs by Equation (3.4).

The round interval of single negotiations was adopted in the experiment to show the efficiency of the proposed approach, which is shown in Figure 3.11.
In Figure 3.11, the horizontal axis indicates single negotiations, and the vertical axis indicates round intervals. Figure 3.11 shows that the values of single negotiations’ round intervals vary from 1 to 60, and the average values of single negotiations’ round intervals are around 30. The experimental result shows a high variation range of round intervals for single negotiations.

Figure 3.12 shows the minimum number of rounds, the average number of rounds, the maximum number of rounds, and the average deadline of agents.

Figure 3.12: Number of rounds and average deadline of agents in single negotiations
In Figure 3.12, the horizontal axis indicates single negotiations, and the vertical axis indicates negotiation rounds. From Figure 3.12, it can be seen that the average values of number of rounds are around 8, and the average values of negotiation deadlines are around 14. Therefore, most of negotiations can successfully reach agreements by spending only 60% of their expected time, and this is a strong evidence to demonstrate the efficiency of the proposed approach.

To summarise, the experimental results show that the proposed approach is effective and able to handle MCN efficiently and effectively.

### 3.5 Case Studies

In this section, three case studies are presented based on following scenarios in MCN, which are a complete success scenario, a partial success scenario and a no success scenario. The MCN in Figure 3.7 and the single-issue negotiation model in Section 3.4 were adopted in the case studies. Other detailed settings and results of the three case studies are as follows.

#### 1. Complete success scenario

In a complete success scenario, all MNRs (i.e., MNRs $R_0, R_1, R_2$) are successful which means that single negotiations (i.e., Negotiations $A_0, A_1, A_2, A_3, A_4, A_5$) achieve success. The parameters for single negotiations are listed in Table 3.1.

**Table 3.1:** Negotiation parameters generated randomly in a complete success scenario

<table>
<thead>
<tr>
<th>Negotiation</th>
<th>Preference</th>
<th>Agent</th>
<th>(Initial Offer, Reserved Offer)</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0$</td>
<td>0.045</td>
<td>$a$</td>
<td>(336k,513k)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_1$</td>
<td>(489k,370k)</td>
<td>17</td>
</tr>
<tr>
<td>$A_1$</td>
<td>0.234</td>
<td>$a$</td>
<td>(322k,506k)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2$</td>
<td>(459k,378k)</td>
<td>11</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0.203</td>
<td>$a$</td>
<td>(331k,502k)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_3$</td>
<td>(451k,383k)</td>
<td>17</td>
</tr>
<tr>
<td>$A_3$</td>
<td>0.077</td>
<td>$a$</td>
<td>(333k,515k)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_4$</td>
<td>(461k,402k)</td>
<td>18</td>
</tr>
<tr>
<td>$A_4$</td>
<td>0.215</td>
<td>$a$</td>
<td>(306k,505k)</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_1$</td>
<td>(470k,392k)</td>
<td>11</td>
</tr>
<tr>
<td>$A_5$</td>
<td>0.226</td>
<td>$a$</td>
<td>(304k,506k)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_4$</td>
<td>(463k,419k)</td>
<td>11</td>
</tr>
</tbody>
</table>

Figure 3.13 and Figure 3.14 show the experimental results of the utilities achieved.
from Negotiations $A_0$, $A_1$, $A_2$, $A_3$, $A_4$, $A_5$, and the utilities achieved from MNRs $R_1$, $R_2$, and MCN $N$, respectively. The utilities achieved from single negotiations are calculated by Equation (3.6).

In Figure 3.13, the horizontal axis shows negotiation time (i.e., in terms of negotiation rounds), and the vertical axis indicates the utility achieved from single negotiations. The curves in Figure 3.13 represent the utilities achieved from single negotiations. Figure 3.13 shows that Negotiation $A_0$ always starts first at Round 0 and Negotiations $A_1$, $A_2$, $A_3$, $A_4$, $A_5$ start at Round 2, Round 4, Round 6, Round 7 and Round 8, respectively. The result shows that all negotiations start and end at different times and are conducted concurrently. Figure 3.13 also shows that utilities achieved from all negotiations increase during the negotiation, and all negotiations achieve success in the end.

In Figure 3.14, the horizontal axis represents negotiation time, and the vertical axis indicates utility. Figure 3.14 shows that the utilities achieved from all MNRs and the MCN increase with time, and each MNR starts and ends at different times. The utility achieved from the MCN (the yellow line in Figure 3.14) does not appear until all negotiations finish their first negotiation rounds at Round 8, and disappears at Round 31 because all negotiations are terminated there. The results from Figure 3.13 and Figure 3.14 show that the proposed approach is able to handle MCN in a
CHAPTER 3. A MULTIPLE CONCURRENT NEGOTIATIONS MODEL

Figure 3.14: Utilities achieved from MNRs in a complete success scenario

complete success scenario.

2. Partial success scenario

In a partial success scenario, only some of MNRs achieve success. In this scenario, MNRs $R_0$, $R_2$ are successful while MNR $R_1$ fails. The parameter settings for single negotiations are listed in Table 3.2.

Table 3.2: Negotiation parameters generated randomly in a partial success scenario

<table>
<thead>
<tr>
<th>Negotiation</th>
<th>Preference</th>
<th>Agent</th>
<th>(Initial Offer, Reserved Offer)</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_0$</td>
<td>0.125</td>
<td>$a$</td>
<td>(312k, 541k)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_1$</td>
<td>(502k, 373k)</td>
<td>16</td>
</tr>
<tr>
<td>$A_1$</td>
<td>0.221</td>
<td>$a$</td>
<td>(362k, 534k)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2$</td>
<td>(480k, 364k)</td>
<td>13</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0.106</td>
<td>$a$</td>
<td>(374k, 511k)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_3$</td>
<td>(454k, 390k)</td>
<td>18</td>
</tr>
<tr>
<td>$A_3$</td>
<td>0.059</td>
<td>$a$</td>
<td>(349k, 505k)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_4$</td>
<td>(450k, 399k)</td>
<td>14</td>
</tr>
<tr>
<td>$A_4$</td>
<td>0.208</td>
<td>$a$</td>
<td>(313k, 517k)</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_4$</td>
<td>(472k, 390k)</td>
<td>14</td>
</tr>
<tr>
<td>$A_5$</td>
<td>0.281</td>
<td>$a$</td>
<td>(325k, 527k)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_4$</td>
<td>(454k, 408k)</td>
<td>12</td>
</tr>
</tbody>
</table>
The utilities achieved from single negotiations are shown in Figure 3.15, and the utilities achieved from the MCN and the MNRs are shown in Figure 3.16.

Figure 3.15: Utilities achieved from single negotiations in a partial success scenario

The curves in Figure 3.15 represent the utilities achieved from single negotiations. From Figure 3.15, it can be seen that each negotiation starts and ends at different times and these negotiations are conducted concurrently. Figure 3.15 also shows that the values of utilities achieved from all negotiations except Negotiation $A_4$ increase as the MCN goes on. Negotiations $A_0, A_1, A_2, A_3, A_5$ achieve success in the end, but Negotiation $A_4$ fails at Round 44.
Figure 3.16: Utilities achieved from MNRs in a partial success scenario

From Figure 3.16, it can be seen that each MNR starts and ends at different times. The utility achieved from MCN N (yellow line in Figure 3.16) appears at Round 10 because all negotiations finish their first negotiation rounds there, and the utility achieved from the MCN disappears at Round 44 since all negotiations finish there. At Round 44, the value of the utility achieved from MNR R_1 declines to 0 due to the failure of Negotiation A_4. Therefore, the value of the utility achieved from MCN N declines at Round 44 as well. In the end, MNRs R_0 and R_2 are successful while MNR R_1 fails. The results in Figure 3.15 and Figure 3.16 show that the proposed approach is able to handle MCN in a partial success scenario.

3. No success scenario

In a no success scenario, all negotiations fail in the end, which means that MCN fails. The parameter settings for single negotiations are listed in Table 3.3.
Table 3.3: Negotiation parameters generated randomly in a no success scenario

<table>
<thead>
<tr>
<th>Negotiation</th>
<th>Preference</th>
<th>Agent</th>
<th>(Initial Offer, Reserved Offer)</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₀</td>
<td>0.033</td>
<td>a</td>
<td>(339k,512k)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₁</td>
<td>(447k,376k)</td>
<td>11</td>
</tr>
<tr>
<td>A₁</td>
<td>0.141</td>
<td>a</td>
<td>(341k,532k)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₂</td>
<td>(481k,407k)</td>
<td>10</td>
</tr>
<tr>
<td>A₂</td>
<td>0.253</td>
<td>a</td>
<td>(325k,539k)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₃</td>
<td>(477k,369k)</td>
<td>15</td>
</tr>
<tr>
<td>A₃</td>
<td>0.207</td>
<td>a</td>
<td>(344k,506k)</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₄</td>
<td>(476k,404k)</td>
<td>11</td>
</tr>
<tr>
<td>A₄</td>
<td>0.179</td>
<td>a</td>
<td>(366k,527k)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₄</td>
<td>(481k,379k)</td>
<td>13</td>
</tr>
<tr>
<td>A₅</td>
<td>0.187</td>
<td>a</td>
<td>(348k,522k)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b₄</td>
<td>(462k,395k)</td>
<td>13</td>
</tr>
</tbody>
</table>

In the no success scenario, the utilities achieved from single negotiations are shown in Figure 3.17, and the utilities achieved from MNRs and MCN are shown in Figure 3.18.

Figure 3.17: Utilities achieved from single negotiations in a no success scenario

The horizontal axis in Figure 3.17 indicates negotiation time, and the vertical axis indicates utility. The curves in Figure 3.17 represent the utilities achieved from single negotiations. Figure 3.17 shows that single negotiations starts and ends at different times. The utilities achieved from Negotiations A₁, A₂, A₃, A₄, A₅ increase
with time. The utility achieved from Negotiation $A_0$ increases at the beginning, then declines to 0 finally due to its failure.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.18}
\caption{Utilities achieved from MNRs in a no success scenario}
\end{figure}

In Figure 3.18, each MNR starts and ends at different times. At Round 10, all negotiations finish their first negotiation rounds, and the value of the utility achieved from MCN $N$ is generated. Similarly, the value of the utility achieved from MCN $N$ disappears at Round 55 because all negotiations finish there. At Round 55, the utilities achieved from MCN $N$, MNRs $R_0$, $R_1$, and $R_2$ decline to 0 due to the failure of Negotiation $A_0$. Therefore, MCN $N$ fails. Since all other negotiations depend on Negotiation $A_0$, and the failure of Negotiation $A_0$ would definitely result in the failure of MCN $N$. The results from Figure 3.17 and Figure 3.18 show that the proposed approach works well in a no success scenario.

\section*{3.6 Summary}

This chapter proposed an MCN model for handling the concurrency in MCN and the issue interdependency across MCN. First, MCN and issue interdependency across MCN were formally defined. The issue interdependencies between multiple negotiations were represented by a directed graph. Then, by applying the proposed algorithms, an MCN’s graph-based representation was converted to its CPN-based representation. Based on an MCN’s CPN-based representation, the MCN was handled by the proposed CPN-based protocol. Additionally, experimental results showed the effectiveness and efficiency of the proposed approach. Therefore, the proposed MCN
model in this chapter could successfully fulfil **Objective 1** of this thesis (refer to Section 1.5).
Chapter 4

An MCN Protocol for a Dynamic Changing Environment

To attain Objective 2 of this thesis (refer to Section 1.5), this chapter proposes a negotiation protocol for handling MCN in a dynamic changing negotiation environment.

A real-world scenario of MCN with a dynamic changing environment is Service Level Agreement (SLA) in service-oriented cloud computing. For instance, a customer might ask for a number of cloud services by processing multiple SLA negotiations. Interdependency relationships exist between these SLA negotiations, where each service’s process somehow impacts the process of other services. In order to maximize a customer’s profit, concurrently processing these interdependent SLA negotiations is the optimal solution. Moreover, in the dynamic cloud computing environment, the customer may change their original requests during the negotiation, i.e., adding new service requests or cancelling ongoing service requests. In such an MCN scenario, any new negotiation could be added and any ongoing negotiation could be removed during the MCN process.

In Chapter 3, an MCN model was proposed for handling MCN in a static negotiation environment, where any change of negotiations is not prohibited during the MCN process. By extending the proposed MCN model in Chapter 3, an MCN protocol is proposed in this chapter for handling MCN in a dynamic changing negotiation environment, where an updating mechanism is newly introduced to handle dynamic changes of negotiations, i.e., any new negotiation can be added and any ongoing negotiation can be removed. During the MCN process, if dynamic changes of negotiations happen, an MCN’s graph-based representation will be firstly updated. Then, the preference distribution on MCN is updated due to the dynamic changes of the negotiations. Based on the MCN’s updated graph-based representation, the MCN’s Colored Petri Net (CPN)-based representation is accordingly updated. Through applying a series of manipulations on the MCN’s CPN-based representation, the
dynamic changes of negotiations are effectively and efficiently handled.

The outline of this chapter is as follows. Section 4.1 proposes an updating mechanism for handling MCN in a dynamic changing negotiation environment. Section 4.2 proposes a negotiation protocol for handling MCN in a dynamic changing negotiation environment. Section 4.3 presents an experiment and the experimental results. Section 4.4 summarises this chapter.

4.1 An Updating Mechanism for Handling Dynamic Changes

This section proposes an updating mechanism for handling dynamic changes in MCN, which includes graph-based representation updating, utility updating and CPN-based representation updating.

4.1.1 Graph-Based Representation Updating

This subsection introduces graph-based representation updating for MCN, which includes the cases of adding negotiations and removing negotiations.

![Figure 4.1: A graph-based representation updating example of changing negotiations](image)

1. Adding negotiations

Figure 4.1(a) shows graph-based representation updating when adding Negotiations $A_3$ and $A_4$. In Figure 4.1(a), Negotiations $A_3$ and $A_4$ are added, and Edge...
(A_0, A_1) is removed as the interdependency relationship of A_0 \preceq A_1 satisfies the property of *transitivity* (refer to Section 3.1 in Chapter 3). Thus, it is not necessary to keep Edge (A_0, A_1) after the creation of Edges (A_0, A_3) and (A_3, A_1).

2. Removing negotiations

Figure 4.1(b) shows graph-based representation updating when removing Negotiations A_3 and A_4. In Figure 4.1(b), Edges (A_0, A_3) and (A_3, A_1) are removed, and Edge (A_0, A_1) is added. Based on the property of *transitivity* (refer to Section 3.1 in Chapter 3), Edge (A_0, A_1) is added to indicate the existing interdependency between Negotiations A_0 and A_1.

### 4.1.2 Utility Updating

According to the calculation of the overall utility achieved from MCN, the overall utility can also be calculated by Equation (4.1) as follows by combining Equation (3.2), Equation (3.4), and Equation (3.5) (refer to Section 3.2.3 in Chapter 3).

$$ U(N) = \sum_{\forall R_i \in N} \left( \sum_{\forall A_j \in R_i} \left( \omega_j \times U(A_j) \times V(A_j) \right) \times \prod_{\forall A_j \in R_i} V(A_j) \right) $$

where \( \omega_j \) indicates the initial preference for Negotiation A_j, and \( \omega'_j \) indicates the updated preference for Negotiation A_j.

Let the set of modified negotiations is \( N_0 = \{ A_{n+p} | 1 \leq p \leq m \} \).

1. Adding negotiations

In MCN \( N = \{ A_1, \cdots A_j, \cdots A_n \} \), preferences for negotiations in the updated MCN \( N' = \{ A_1, \cdots A_j, \cdots A_n, A_{n+1}, \cdots, A_{n+m} \} \) are calculated by Equation (4.2) as follows.

$$ \omega'_j = \frac{\omega_j}{1 + \sum_{p=1}^{m} \omega_{n+p}} $$

where \( \omega_j \) indicates the initial preference for Negotiation A_j, and \( \omega'_j \) indicates the updated preference for Negotiation A_j.

2. Removing negotiations

In MCN \( N = \{ A_1, \cdots A_j, \cdots A_n, A_{n+1}, \cdots A_{n+m} \} \), preferences for negotiations in the
updated MCN $N' = \{A_1, \cdots A_j, \cdots A_n\}$ are calculated by Equation (4.3) as follows.

$$\omega'_j = \frac{\omega_j}{1 - \sum_{p=1}^{m} \omega_{n+p}}$$

(4.3)

where $\omega_j$ indicates the initial preference for Negotiation $A_j$, and $\omega'_j$ indicates the updated preference for Negotiation $A_j$.

### 4.1.3 CPN-Based Representation Updating

In the MCN’s Colored Petri Net (CPN)-based representation, transitions represent single negotiations, and places represent states of single negotiations. The inputs and outputs of single negotiations are shown by arc directions, and Token $(A, m)(m \geq 1)$ indicates that one offer is received in the $m$th negotiation round in Negotiation $A$ (i.e., enable Transition $t_A$). The first place and last place of each MNR are the initial place and the final place, respectively. Each initial place contains at least one token to fire following transitions. MNR $R_i$ is the $i$th MNR (refer to Definition 3.1.3) in MCN $N$.

Figure 4.2: The CPN-based representation of an MNR

Figure 4.2 shows the CPN-based representation of MNR $R_i = \{A_0, A_1, \cdots A_j, \cdots A_{k-1}\}(0 \leq j \leq k-1)$. Token $(R, m)$ in the initial place (i.e., Place $P_0$) is used to activate MNR $R_i$. If Transition $t_{A_j}$ is activated by a token, it means that Negotiation $A_j$ finishes a negotiation round. If Negotiation $A_j$ is failed, Transition $t_{A_j}$ will not be activated, and the conduct of MNR $R_i$ will be terminated. If Negotiation $A_j$ succeeds, the token will fire the following transitions in MNR $R_i$ based on the arc directions. If all transitions in MNR $R_i$ have been activated, it is called that $R_i$ finishes an MNR negotiation round. In order to handle the concurrency in MCN, a backward arc (i.e., from Transition $t_{A_0}$ to Place $P_0$) is added to show that different negotiations are performed concurrently.

The CPN-based representation of MCN is transferred from its corresponding graph-based representation. Therefore, the mechanism of CPN-based representation updating is similar to the graph-based representation updating mechanism. The following example is taken to show the CPN-based representation updating for MCN.

Figure 4.3 shows the graph-based representation updating of MCN, where the
initial MCN is $N = \{A_0, A_1, \ldots A_{k-2}\}$ and the update MCN is $N' = \{A_0, A_1, \ldots A_{k-2}, A_{k-1}\}$. It can be seen that Negotiation $A_{k-1}$ is added after Negotiation $A_{k-2}$.

![Figure 4.3: An example of the graph-based representation updating for MCN](image1)

![Figure 4.4: An example of the CPN-based representation updating for the MCN](image2)

Figure 4.4 shows the according CPN-based representation updating for the MCN, whose graph-based representation is shown in Figure 4.3(a). From Figure 4.4(b), it can be seen that Transition $t_{A_{k-1}}$, Place $P_k$ and the arc $(t_{A_{k-1}}, P_k)$ are added after Place $P_{k-1}$.

In the graph-based representation of MCN, a negotiation is represented by a node, and the issue interdependency is represented by a directed edge. In the CPN-based representation, a negotiation is represented by a transition, and the issue
interdependency is represented by an arc. Therefore, updating a CPN-based representation obeys similar rules as updating its graph-based representation. The only difference is that changing transitions in a CPN-based representation accompanies with changing corresponding places.

4.2 A Negotiation Protocol for MCN in a Dynamic Changing Environment

In this section, a negotiation protocol for processing MCN in a dynamic changing environment is proposed.

The inputs of Algorithm 6 are MCN $N$ and the updated MCN $N'$, and the

<table>
<thead>
<tr>
<th>Algorithm 6 A negotiation protocol for CMN in a dynamic changing environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> MCN $N$, the updated MCN $N'$</td>
</tr>
<tr>
<td><strong>Output:</strong> the overall utility from the MCN, and the result of conducting the MCN, i.e., success or failure.</td>
</tr>
<tr>
<td>1: Generate MCN $N$'s graph-based representation according to the issue interdependency;</td>
</tr>
<tr>
<td>2: Generate MCN $N$'s corresponding CPN-based representation $C_N$;</td>
</tr>
<tr>
<td>3: Start conducting MCN $N$;</td>
</tr>
<tr>
<td>4: while MCN $N$ is not completed do</td>
</tr>
<tr>
<td>5: Keep conducting MCN $N$;</td>
</tr>
<tr>
<td>6: if negotiations are requested to be added or removed then</td>
</tr>
<tr>
<td>7: Execute updating the graph-based representation and the CPN-based representation of MCN $N$ to get the updated CPN-based representation $C_{N'}$;</td>
</tr>
<tr>
<td>8: Keep conducting the updated MCN $N'$ according to the updated CPN-based representation $C_{N'}$;</td>
</tr>
<tr>
<td>9: while every negotiation finishes a negotiation round do</td>
</tr>
<tr>
<td>10: calculate $V(N')$ and $U(N')$ by employing the proposed updating mechanism;</td>
</tr>
<tr>
<td>11: if $V(N') &lt; \Omega_{N'}$ then</td>
</tr>
<tr>
<td>12: keep executing the updated MCN $N'$;</td>
</tr>
<tr>
<td>13: else if $V(N') \geq \Omega_{N'}$ then</td>
</tr>
<tr>
<td>14: terminate MCN $N'$ and quit;</td>
</tr>
<tr>
<td>15: return $U(N')$ and success;</td>
</tr>
<tr>
<td>16: end if;</td>
</tr>
<tr>
<td>17: end while;</td>
</tr>
<tr>
<td>18: else if no negotiations are requested to be changed then</td>
</tr>
<tr>
<td>19: $C_{N'} = C_N$;</td>
</tr>
<tr>
<td>20: end if;</td>
</tr>
<tr>
<td>21: end while;</td>
</tr>
<tr>
<td>22: if $V(N) &lt; \Omega_N$ or $V(N') &lt; \Omega_{N'}$ then</td>
</tr>
<tr>
<td>23: return failure;</td>
</tr>
<tr>
<td>24: end if.</td>
</tr>
</tbody>
</table>
outputs of the algorithm are the overall utility achieved from MCN and the result of conducting the MCN (i.e., success or failure). At the beginning, according to the issue interdependency between negotiations, the algorithm generates the MCN’s graph-based representation and its corresponding CPN-based representation $C_N$, and then starts conducting the MCN (Lines 1-3). If dynamic changes happen, the algorithm executes the proposed updating mechanism to get the updated CPN-based representation $C_{N'}$. Then, the algorithm keeps conducting the updated MCN $N'$ based on the updated CPN-based representation $C_{N'}$ (Lines 6-8). If every negotiation finishes a negotiation round, the algorithm computes the updated MCN $N'$'s success possibility (i.e., $V(N')$) and the utility achieved from the updated MCN $N'$ (i.e., $U(N')$) (Lines 9-10). If the value of $V(N')$ is less than the value of MNG $\Omega_N$, the algorithm keeps conducting the updated MCN $N'$. Otherwise, the algorithm terminates the updated MCN $N'$ and quits. The algorithm shows that if the agent’s multi-negotiation goal $\Omega_N$ (refer to Section 3.1) is achieved, it returns the overall utility achieved from the updated MCN (i.e., $U(N')$) and “success” as the outputs (Lines 11-16). Because it is not necessary to execute other unfinished negotiations, it can improve efficiency in some extent. If no negotiation is required to be changed during the process of negotiations, the algorithm keeps conducting MCN $N$ according to its original CPN-based representation $C_N$ and compares the values of $V(N)$ and $U(N)$ to decide whether to keep conducting the MCN or to terminate it (Lines 18-20). If the algorithm completely finishes conducting the MCN and the multi-negotiation goal is not achieved, the algorithm returns “failure” as the output (Lines 22-24).

4.3 Experiment

In the experiment, an example of MCN is taken as a basis (see Figure 4.5). Based on the structure of the MCN, all possible positions are considered for adding and removing negotiations during the process of the MCN. Specifically, three different dynamic changing scenarios are considered in this work, which are adding negotiations, removing negotiations, and simultaneously adding and removing negotiations. The detailed experimental settings are described as follows.

4.3.1 Experimental Settings for Static Negotiations

In the experimental settings, a basic MCN is taken, where $N = \{A_0, A_1, A_2, A_3\}$. There are two MNRs (refer to Definition 3.1.3 in Chapter 3) in Figure 4.5, which are $R_0 = \{A_0, A_1, A_2\}$ and $R_1 = \{A_0, A_1, A_3\}$.

In the experiment, a single-issue negotiation model [FWJ02] is employed to conduct each negotiation, where the utility function for every single negotiation is
described as Equation (4.4).

\[
U(\text{counter offer}) = \frac{\text{reserved offer} - \text{counter offer}}{\text{reserved offer} - \text{initial offer}} \tag{4.4}
\]

The parameters for each negotiation are described in Table 4.1. The preferences for negotiations are selected randomly, and concession strategies for negotiations are randomly picked up from Conceder, Linear and Boulware strategies [FSJ98].

![Figure 4.5: The basic MCN](image)

The parameters for each negotiation are described in Table 4.1. The preferences for negotiations are selected randomly, and concession strategies for negotiations are randomly picked up from Conceder, Linear and Boulware strategies [FSJ98].

**Table 4.1: Parameters in single negotiations**

<table>
<thead>
<tr>
<th></th>
<th>initial offer</th>
<th>reserved offer</th>
<th>concession strategy</th>
<th>deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>random from [300k, 350k]</td>
<td>random from [450k, 500k]</td>
<td>random from {(0, 1), (1, 5)}</td>
<td>random from [10, 20]</td>
</tr>
<tr>
<td>opponents</td>
<td>random from [500k, 550k]</td>
<td>random from [370k, 420k]</td>
<td>random from {(0, 1), (1, 5)}</td>
<td>random from [10, 20]</td>
</tr>
</tbody>
</table>

**4.3.2 Experimental Settings for Dynamic Negotiations**

To get general results of the proposed protocol’s performance, the mandatory overall goal is not specified for an agent. The MNG (refer to Section 3.1 in Chapter 3) indicates the expected outcome of conducting MCN. In the experimental settings, an agent’s goal is classified into two intervals, i.e., MNG = \{[1/2, 1), 1\}. “MNG = 1” indicates that the expected outcomes of all MNRs are successful, and “MNG = [1/2, 1)” indicates that at least 50% all involved MNRs but not all MNRs reach successful outcomes.

To better test the proposed protocol in a dynamic changing negotiation environment, all possible positions of adding and removing negotiations are considered in the experiment (see Figure 4.6). Static negotiations (i.e., from Negotiations $A_0$ to $A_3$) are shown by bold circles, and modified negotiations (i.e., from Negotiations $A_4$ to $A_{14}$) are shown by dashed circles. For simplification, the dynamic changes of
all cases in the following three scenarios happen in the same negotiation round.

1. Adding negotiations

Let a negotiation set $N_{\text{add}}$ indicate all cases of adding negotiations, where $N_{\text{add}} = \{N'_a | N'_a \subseteq N_a, N'_a \neq \emptyset\}$ and $N_a = \{A_i | i \in [4, 14]\}$.

2. Removing negotiations

Let a negotiation set $N_{\text{remove}}$ indicate all cases of removing negotiations, where $N_{\text{remove}} = \{N'_b | N'_b \subseteq N_b, N'_b \neq \emptyset\}$ and $N_b = \{A_i | i \in [1, 3]\}$. There are two special cases in this scenario. The first one is removing Negotiation $A_0$. Due to the issue interdependency, all other negotiations will be removed if removing Negotiation $A_0$. The second case is simultaneously removing Negotiations $A_1$, $A_2$, and $A_3$. There will be only Negotiation $A_0$ left in this case, and it can be treated as a single negotiation. Therefore, these two special cases are not considered.

3. Simultaneously adding and removing negotiations

Let a negotiation set $N_{\text{mix}}$ indicate all cases of simultaneously adding and removing negotiations, where $N_{\text{mix}} = N_{\text{add}} \times N_{\text{remove}}$. Some cases in this scenario do not exist due to the experimental settings and the graph-based representation in Figure 4.6. The details of these special cases are explained as follows.

- If the number of removed negotiations is 1, the maximum number of added negotiations is 9;
• If the number of removed negotiations is 2, the maximum number of added negotiations is 7;
• If the number of removed negotiations is 3, the maximum number of added negotiations is 3.

4.3.3 Experimental Results

Based on experimental settings, the average percentage of achieving an agent’s MNG and the overall utility achieved from MCN are tested in three dynamic scenarios, respectively. Here, the average percentage is taken as a result since there are many cases of randomly selecting adding/removing negotiations. Moreover, some special cases are conducted 100 times (i.e., the static case, adding eleven negotiations, simultaneously adding three and removing three negotiations). The reason is that the selection of adding or removing negotiations in each special case is unique. Black vertical lines are introduced to indicate the deviations of utilities achieved from MCN.

1. Adding negotiations

Figure 4.7: Percentage of an agent’s MNG in the scenario of adding negotiations

Figure 4.7 shows the average percentage of achieving an agent’s MNG, where
MNG = \{[1/2, 1), 1\}. In Figure 4.7, the x-axis shows all cases of adding negotiations (i.e., “+3” indicates adding three negotiations), and the y-axis indicates the average percentage of achieving the MNG. The average percentage of achieving “MNG = [1/2, 1)” goes up when adding more negotiations. For “MNG = 1”, the average percentage of achieving it goes down when adding more negotiations. The reason is that adding more negotiations would increase the possibility of failed negotiations.

Figure 4.8: Overall utility from negotiations in the scenario of adding negotiations

Figure 4.8 shows the overall utility achieved from the MCN in all cases of adding negotiations. The x-axis indicates all cases of adding negotiations, and the y-axis indicates the overall utility achieved from the MCN. Figure 4.8 shows that, with adding more negotiations, the overall utility achieved from the MCN decreases. The reason is that more negotiations would make it hard to get a higher overall utility achieved from the MCN. However, the overall utility tends to be relatively steady when adding more than four negotiations, which indicates that the proposed protocol well handles the scenario of adding negotiations.
2. Removing negotiations

Figure 4.9: Percentage of an agent’s MNG in the scenario of removing negotiations

Figure 4.10: Overall utility from negotiations in the scenario of removing negotiations
CHAPTER 4. AN MCN PROTOCOL FOR A DYNAMIC CHANGING ENVIRONMENT

Figure 4.9 shows the average percentage of achieving the MNG in the scenario of removing negotiations. The x-axis and the y-axis indicate all cases in this scenario (i.e., “-1” indicates removing one negotiation) and the average percentage of achieving the MNG, respectively. Figure 4.9 shows that when removing more negotiations, the average percentages of achieving “MNG = 1” and “MNG = [1/2, 1)” go up and down, respectively. The reason is that removing negotiations would decrease the number of failed negotiations, and having fewer failed negotiations can increase the possibility of achieving “MNG = 1”. In the case of removing two negotiations, the average percentage of achieving “MNG = [1/2, 1)” is 0, because there is only one MNR left in this case.

Figure 4.10 shows the overall utility achieved from the MCN in the scenario of removing negotiations. The x-axis indicates all cases of removing negotiations, and the y-axis indicates the overall utility achieved from the MCN. It can be seen that, when removing more negotiations, the overall utility achieved from the MCN slightly increases, because it is easier to make an agreement with fewer negotiations. The results also show that the proposed protocol works well in the scenario of removing negotiations.

3. Simultaneously adding and removing negotiations

Figure 4.11: Percentage of an agent’s MNG in the scenario of simultaneously adding and removing negotiations

Figure 4.11 shows the average percentage of achieving the MNG in the scenario of simultaneously adding and removing negotiations. The x-axis indicates all cases
in this scenario (i.e., “-2+3” indicates simultaneously adding three and removing two negotiations), and the y-axis indicates the average percentage of achieving the MNG. From Figure 4.11, it can be seen that if the number of removed negotiations is fixed, adding more negotiations can decrease the average percentage of achieving “MNG = 1” and increase the average percentage of achieving “MNG = [1/2, 1)”. If dividing all data bars in Figure 4.11 into three parts based on the number of removed negotiations, it can be seen that removing more negotiations can make the average percentage of achieving “MNG = 1” slightly go up and make the average percentage of achieving “MNG = [1/2, 1)” go down, respectively. These results have a good match with the results in both scenarios of adding negotiations and removing negotiations. In the special case of “-3+1”, the average percentage of achieving “MNG = [1/2, 1)” is 0, because there is only one MNR left.

Figure 4.12 shows the overall utility achieved from the MCN in the scenario of simultaneously adding and removing negotiations. The x-axis and y-axis indicate all cases in this scenario and the overall utility achieved from the MCN, respectively. From Figure 4.12, it can be seen that if the number of removed negotiations is fixed, the overall utility achieved from the MCN can decline when adding more negotiations. If dividing all data bars in Figure 4.12 into three parts based on the number of removed negotiations, the overall utility achieved from the MCN would slightly go up when removing more negotiations. These results match the results of the overall utility achieved from the MCN in both scenarios of adding negotiations and removing negotiations. The results show that the proposed protocol
well handles the dynamic changes of negotiations when adding negotiations and removing negotiations happen simultaneously.

In summary, the experimental results show that: (1) the proposed protocol is effective while considering all possible changes in three dynamic scenarios, (2) when dynamic changes happen, the proposed protocol is able to handle the concurrency in MCN as well as issue interdependency across MCN.

4.4 Summary

This chapter proposed a negotiation protocol for handling MCN in a dynamic changing negotiation environment. By extending the MCN model proposed in Chapter 3, an updating mechanism was newly introduced in this chapter to effectively handle dynamic changes of negotiations during the MCN process. An experiment was conducted to test the performance of the proposed MCN protocol, where all possible positions of adding and removing negotiations were considered. The experimental results showed that the proposed MCN protocol in this chapter could successfully fulfil Objective 2 of this thesis (refer to Section 1.5).
Chapter 5

Three Negotiation Procedures for MCN

To achieve Objective 3 of this thesis (refer to Section 1.5), this chapter proposes three feasible negotiation procedures for conducting MCN. First, a general MCN model is proposed, where multiple multi-issue negotiations are involved in MCN. Second, an approximating Pareto-optimal offer generation strategy is introduced. In the end, three feasible negotiation procedures for conducting MCN are proposed, i.e., the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. Additionally, the analysis of the three proposed negotiation procedures is also given based on experimental results.

Chapters 3 and 4 proposed an MCN model and an MCN protocol. However, those two chapters focused on MCN with single-issue negotiations. To effectively handle multi-issue negotiation scenarios, this chapter proposes solutions for MCN with multi-issue negotiations.

The outline of this chapter is as follows. Section 5.1 presents a general MCN model. Section 5.2 proposes a negotiation strategy for generating approximating Pareto-optimal offers while considering issue interdependency across MCN. Section 5.3 proposes three negotiation procedures for conducting MCN. Section 5.4 introduces an experimental analysis of the proposed negotiation procedures. Section 5.5 summarises this chapter.

5.1 A General Model for MCN

This section proposes a general model for MCN. The MCN conducted by an agent can form a negotiation network. In this negotiation network, the agent negotiates with different opponents in different negotiations, and each negotiation contains different numbers of negotiation issues. Such a negotiation network for an agent is
formalised as follows.

Let $\Theta = \{N, A, S, O, I\}$ be a negotiation network in which Agent $a$ conduct MCN. In details, let

- $N = \{N_0, \cdots N_i, \cdots N_m\} (i \geq 0)$ denote a set of negotiations involved in the MCN, where $N_i$ indicates a bilateral single-issue/multi-issue negotiation;

- $A = \{a_{N_0}, \cdots a_{N_i}, \cdots a_{N_m}\} (i \geq 0)$ denote a set of Agent $a$’s negotiation opponents in single negotiations, where $a_{N_i}$ is Agent $a$’s opponent in Negotiation $N_i$;

- $S = \{S_{N_0}, \cdots S_{N_i}, \cdots S_{N_m}\} (i \geq 0)$ denote a set of issue sets for every single negotiation, where $S_{N_i} = \{s_0, \cdots s_j, \cdots s_n\} (j \geq 0)$ denotes all issues included in Negotiation $N_i$;

- $O = \{O_{N_0}, \cdots O_{N_i}, \cdots O_{N_m}\} (i \geq 0)$ denote a set of offer sets for all negotiations, where $O_{N_i} = \{O_{a_{N_i}}, O_{a_{N_i}}\}$ indicates a set of offer sets for Negotiation $N_i$. $O_{a_{N_i}} = \{o_{a_{N_i}}, o_{a_{N_i}}^b, \cdots o_{a_{N_i}}^{h}\}$ is a set of offers proposed by Agent $a$ in Negotiation $N_i$, and $O_{a_{N_i}} = \{o_{a_{N_i}}, o_{a_{N_i}}^b, \cdots o_{a_{N_i}}^{h}\}$ is a set of counter-offers proposed by Opponent $a_{N_i}$ in Negotiation $N_i$, e.g., $o_{a_{N_i}}$ indicates a counter-offer given by Opponent $a_{N_i}$ in Negotiation $N_i$ at period $t_i$;

- $I = \{i_0, \cdots i_p, \cdots i_q\}$ indicate a set of issue interdependencies, where $i_p = \{s_l \rightsquigarrow \cdots \rightsquigarrow s_m \rightsquigarrow \cdots \rightsquigarrow s_n\} \in S_{N_l} \times \cdots \times S_{N_j} \times \cdots \times S_{N_k}$ indicates a set of issues satisfying an issue interdependency defined by Definition 5.1.1.

In the proposed MCN model, every single Negotiation $N_i$ is defined as a bilateral negotiation, i.e., a negotiation involves exactly two agents. A conventional multi-lateral negotiation, i.e., a negotiation involves multiple agents, is represented by multiple bilateral negotiations with the exactly same negotiation settings. Therefore, the proposed MCN model is general and can cover all possible negotiation scenarios.

To handle MCN with multi-issue negotiations, the formal definition of issue interdependency across MCN in this Chapter is extended from the issue interdependency defined in Chapter 3, and it is introduced as follows.

**Definition 5.1.1. (Issue Interdependency)** Let $i_p = \{s_l \rightsquigarrow \cdots \rightsquigarrow s_m \rightsquigarrow \cdots \rightsquigarrow s_n\} \in S_{N_l} \times \cdots \times S_{N_j} \times \cdots \times S_{N_k}$ denote an issue interdependency between multiple issues in MCN, and it reflects an interactive restriction between issues across multiple negotiations. Specifically, “$s_l \rightsquigarrow s_m$” indicates that the sub-offer on Issue $s_l$ in Negotiation $N_l$ and the sub-offer on Issue $s_m$ in Negotiation $N_j$ are influenced with each other, and $s_l \rightsquigarrow s_m \iff s_m \rightsquigarrow s_l$. 
Let $o'_a \in O_a(i.e., O_a \in \mathbb{F}_{N_j}, N_j \in \mathbb{N})$ denote the offer given by Agent $a$ at negotiation period $t$ in Negotiation $N_j$, and $o'_a(s)$ ($s \in S_{N_j}$) indicates the sub-offer on Issue $s$. Assuming an issue interdependency is $s_i \sim \cdots \sim s_m \sim \cdots \sim s_n$, Equation (5.1) represents the mathematical relationship between issues in an issue interdependency.

$$\sum_{m=t}^{n} (\mu_m \times o'_{N_j}(s_m)) \leq Q, \quad (5.1)$$

where $\mu_m$ is the coefficient for the sub-offer on Issue $s_m$ in Negotiation $N_j$, $Q$ is a constant, and $t_k$ indicates a negotiation period.

### 5.2 An Offer Generation Strategy in MCN

Agent $a$’s strategy of generating offers in single negotiation with multiple issues is not the focus of this thesis. Therefore, the “shortest distance strategy” [LSL07] is adopted and an extension is done to handle issue interdependencies across MCN in this work. The reason why this strategy is employed is that the “shortest distance strategy” has been proved to be a strategy which generates an approximating Pareto-optimal solution in multi-issue negotiation [LSL07, LSL08].

The core idea of the “shortest distance strategy” is to always select the point which has the shortest distance with the point on its opponent’s indifference curves (surfaces) (i.e., points on an agent’s indifference curves (surfaces) denote the same utility for the agent).

To better describe an agent’s strategy affected by issue interdependency across MCN, “Edgeworth-Bowley Box” [Sch08] is employed, which is frequently utilised in “equilibrium theory” and it aids in bargaining problems of game theory.

To simplify the discussion, a simple negotiation scenario is taken as an example, where Agent $a$ conducts MCN with its opponents. One of the negotiations in the MCN is $N_0$, Agent $a$’s opponent in Negotiation $N_0$ is $a_{N_0}$, the negotiation issues are $s_0$ and $s_1$. Issue $s_1$ may have an interdependency with issues in other negotiations, e.g., $i_0 = \{s_1 \sim \cdots \sim s_m \sim \cdots \sim s_n \mid (s_1, \cdots, s_m, \cdots, s_n) \in S_{N_0} \times \cdots \times S_{N_j} \times \cdots \times S_{N_k}\}$.

Figure 5.1 shows Agent $a$’s strategy of generating offers while considering the issue interdependency $i_0$.

At period $t_0 - 1$ (see Figure 5.1), Agent $a$’s opponent $a_{N_0}$ proposes a counter-offer $o'_{a_{N_0}}$ to Agent $a$. At period $t_0$, if Issue $s_1$ is not involved in the issue interdependency $i_0$, Agent $a$ selects an offer $o'_a$, represented by a black square, which has the shortest distance with point $o'_{a_{N_0}}$. However, if Issue $s_1$ is involved in the issue interdependency $i_0$, Agent $a$ selects an offer $\tilde{o}'_a$, represented by a black dot, which simultaneously satisfies conditions of having the shortest distance with point $o'_{a_{N_0}}$ and falling in the shaded area $P_{a_{t_0}}$, where the shaded area $P_{a_{t_0}}$ is calculated by the
mathematical restriction of the issue interdependency $i_0$ between Issue $s_1$ and other issues involved in the issue interdependency $i_0$. The values of the sub-offers on these other issues are the last values offered in their corresponding negotiations. Based on these values and the mathematical restriction of the issue interdependency $i_0$, the value range of the sub-offer on Issue $s_1$ can be calculated, then the shaded area $P_{a_0}^{t_0}$ is determined. By employing the proposed strategy, the negotiation is conducted until an offer is accepted or a deadline is reached.

### 5.3 Negotiation Procedures for MCN

The negotiation procedure for MCN indicates how agents conduct MCN, which is crucial to the success of conducting MCN. Let $N = \{N_0, \ldots, N_i, \ldots, N_m\}$ be the MCN conducted by Agent $a$, where $N_i$ denotes a bilateral single-issue/multi-issue negotiation. It is assumed that Agent $a$ has its private preference for the importance of each negotiation in the MCN, which is represented by a set $V = \{\nu_0, \ldots, \nu_i, \ldots, \nu_m\}$, where $\nu_i$ indicates the importance of negotiation $N_i$ for Agent $a$. This section proposes the following three negotiation procedures to conduct MCN in different situations.

#### 1. Concurrent Negotiation Procedure

In the concurrent negotiation procedure, the agent concurrently processes all negotiations. The agent’s decision-making in each negotiation is affected by issue interdependency across MCN during the negotiation process.
As very little work exits in solving the problem of achieving the concurrency in MCN, this chapter proposes the following transition system-based approach to solve this problem.

To handle the concurrency in MCN, the transition system is employed in the concurrent negotiation procedure.

A general transition system $TS$ is a tuple $TS = < S, I, Act, G >$ [BKL08], where $S$ indicates a set of states, $I$ indicates a set of initial states and $I \subseteq S$. $Act = \{a_0, \cdots a_i, \cdots a_c\}$ is a set of actions, where $a_i \in S \times S$ and $G$ is a set of final states and $G \subseteq S$.

Based on the definition of the general transition system, a transition system-based representation of multiple negotiations is defined as follows.

A set of negotiations $N = \{N_0, \cdots N_i, \cdots N_m\}$ in a negotiation network $\Theta_a$ is represented by a concurrent system $TS_N = TS_{N_0} || \cdots ||TS_{N_i} || \cdots ||TS_{N_m}$, where $TS_{N_i}$ indicates a transition system-based representation of Negotiation $N_i$, i.e., $TS_{N_i} = < S, I, Act, G, cs >$, where $S = \{\text{initial, ongoing, failure, success}\}$, $I = \{\text{initial}\}$, $Act = \{(\text{initial, ongoing)}, (\text{ongoing, ongoing}), (\text{ongoing, failure}), (\text{ongoing, success})\}$, $G = \{\text{failure, success}\}$, and $cs \in S$ indicates the current state of Negotiation $N_i$.

Algorithm 7 shows the concurrent negotiation procedure for conducting MCN, where the concurrency in MCN is handled by the proposed transition system-based approach.

Algorithm 7 Concurrent Negotiation Procedure of MCN

Input: a negotiation network $\Theta_a$, which involves MCN.
Output: the outcomes of all negotiations in the MCN.

1: Pre-calculations: calculate the transition system $TS_N$.
2: concurrently start conducting all $m$ negotiations in MCN $N = \{N_0, \cdots N_i, \cdots N_m\}$
3: while not all negotiations are completed do
4: if Agent $a$ receives a counter-offer from its opponent in Negotiation $N_i$ then
5: retrieve all the last values of sub-offers on issues in other negotiations which have interdependencies with issues in Negotiation $N_i$, and store the values in a list $lis$
6: end if;
7: adopt the proposed strategy in Section 5.2 while considering the values in the list $lis$
8: if Negotiation $N_i$ is completed then
9: record the result of Negotiation $N_0$, i.e., success or failure, and calculate the utility achieved from Negotiation $N_i$, i.e., $U(N_i)$
10: end if;
11: end while;
12: return the outcomes of all negotiations.

In Algorithm 7, the input is a negotiation network $\Theta_a$ involving MCN, and the output is the outcomes of all negotiations in the MCN. Firstly, the transition system $TS_N$ is precalculated based on the negotiation network $\Theta_a$ (Line 1). Then, Agent $a$ concurrently starts conducting all negotiations in the MCN (Line 2). During the
negotiations, if Agent $a$ receives a counter-offer from its opponent in Negotiation $N_i$, Agent $a$ checks issue interdependency and retrieves all the last values of sub-offers on the related issues in other negotiations (Lines 4-5). Agent $a$ adopts the proposed strategy in Section 5.2 to negotiate with its opponent (Line 7). For each completed negotiation, Agent $a$ records the result of the negotiation and calculates the utility achieved from the negotiation (Lines 8-9). All $m$ negotiations are concurrently conducted until all of them are completed.

2. Successive Negotiation Procedure

In some negotiation scenarios, an agent might prefer to conduct multiple negotiations according to the importance of every single negotiation, which means the agent always processes the most important negotiation. Under this consideration, the successive negotiation procedure is proposed.

In the successive negotiation procedure, an agent processes multiple negotiations one after another. The sequence of processing these negotiations is dependent on their importance. After the former negotiation is completed, the agent processes the next one. Once a negotiation is completed, all issues in the completed negotiation have been settled. The agent’s decision-making in the latter negotiations is affected by interdependencies from settled issues in completed negotiations.

Algorithm 8 shows the successive negotiation procedure for processing MCN conducted by Agent $a$.

**Algorithm 8** Successive Negotiation Procedure of MCN

**Input:** a negotiation network $\Theta_a$, which involves MCN.

**Output:** the outcomes of all negotiations in the MCN.

1. **Pre-calculations:** sort all $m$ negotiations in the MCN based on their importance $\nu$ and get the negotiation sequence, e.g., $N_0 \rightarrow \cdots \rightarrow N_i \rightarrow \cdots \rightarrow N_m$, where $\nu_0 \geq \cdots \geq \nu_i \geq \cdots \geq \nu_m$.
2. for $i \leftarrow 0$ to $m$ do
3. if issues in Negotiation $N_i$ have issue interdependencies with issues in other negotiations then
4. retrieve all the latest values of sub-offers on issues related in issue interdependencies and store the values in a list $lis$
5. end if;
6. conduct Negotiation $N_i$ by using the proposed strategy in Section 5.2 while considering the values in the list $lis$
7. if Negotiation $N_i$ is completed then
8. record the result of Negotiation $N_0$, i.e., success or failure, and calculate the utility achieved from Negotiation $N_i$, i.e., $U(N_i)$
9. end if;
10. $i \leftarrow i + 1$
11. end for;
12. return the outcomes of all negotiations.
In Algorithm 8, the input is a negotiation network $\Theta_a$ involving MCN, and the output is outcomes of all negotiations in the MCN. Firstly, Agent $a$ sorts all $m$ negotiations based on their importance, and gets a negotiation sequence, e.g., $N_0 \rightarrow \cdots \rightarrow N_i \rightarrow \cdots \rightarrow N_m$ (Line 1). Then Agent $a$ conducts all negotiations based on the negotiation sequence. During conducting a negotiation (e.g., $N_i$), the agent checks interdependencies of issues in Negotiation $N_i$, then retrieves all the last values of sub-offers on the issues, where these issues have interdependencies with the issues in Negotiation $N_i$ (Lines 3-4). Agent $a$ adopts the proposed strategy in Section 5.2 to negotiate with its opponent (Line 6). For each completed negotiation, Agent $a$ records the result of the negotiation and calculates the utility achieved from the negotiation (Lines 7-8). Agent $a$ follows the same procedure to successively conduct all $m$ negotiations until all of them are completed.

3. Clustered Negotiation Procedure

In some negotiation scenarios, an agent might prefer to bundle negotiations as multiple packages due to the issue interdependency and the importance of every single negotiation, and to conduct the negotiations in multiple packages according to the importance of the negotiations involved in the packages. Under this consideration, the clustered negotiation procedure is proposed.

In the clustered negotiation procedure, all $m$ negotiations are firstly partitioned into $\mu > 1$ disjoint subsets, where each subset is called a *negotiation cluster* in this chapter. The negotiations in each negotiation cluster are processed by using the *concurrent negotiation procedure*. Then, Agent $a$ processes all negotiation clusters

![Figure 5.2: An example of issue interdependency across MCN](image)
by using the *successive negotiation procedure*. Once a negotiation cluster is completed (i.e., all negotiations in the negotiation cluster are completed), all issues in the completed negotiation cluster have been settled, and the agent’s decision-making in latter negotiation clusters is affected by the issue interdependencies from the completed negotiation clusters. In general, the *clustered negotiation procedure* covers both the *successive negotiation procedure* and the *concurrent negotiation procedure*. For example, when $\mu = m$, there are $m$ negotiation clusters which turns the *clustered negotiation procedure* to the *successive negotiation procedure*; when $\mu = 1$, there is only one negotiation cluster, which turns the *clustered negotiation procedure* to the *concurrent negotiation procedure*. The detailed explanation of how to cluster MCN is presented as follows.

Figure 5.2 shows an example of issue interdependency across MCN. Each big circle indicates a negotiation with multiple issues, where each black dot indicates a negotiation issue, an edge between two black dots indicates an issue interdependency across MCN, and no edge between black dots indicates no issue interdependency across MCN. This example shows that there are four multi-issue negotiations in the MCN, which are Negotiations $N_0$, $N_1$, $N_2$, $N_3$, and there are three issue interdependencies across MCN, which are $s_{00} \sim s_{20} \sim s_{30}$, $s_{01} \sim s_{31}$ and $s_{10} \sim s_{21}$.

For partitioning multiple negotiations into clusters, the idea of spectral clustering [NJW02, Lux07] is borrowed. In this work, the “distance” between two negotiations is relevant to the *negotiation strength* between the two negotiations. The *negotiation strength* $\xi_{ij}$ between Negotiations $N_i$ and $N_j$ is calculated by the number of edges between Negotiations $N_i$ and $N_j$. For instance, $\xi_{01} = \xi_{13} = 0$, $\xi_{02} = \xi_{12} = \xi_{23} = 1$, and $\xi_{03} = 2$.

Algorithm 9 shows the clustered negotiation procedure for conducting MCN.

In Algorithm 9, the input is a negotiation network $\Theta_a$ involving MCN, and the output is the outcomes of all involved negotiations. The pre-calculations include: (1) Agent $a$ calculates the negotiation strength between every two negotiations, (2) Agent $a$ applies a clustering algorithm to partition all $m$ negotiations into $k$ negotiation clusters, and (3) Agent $a$ gets the cluster sequence by sorting all negotiation clusters through calculating the importance of negotiation clusters, i.e., the sum of importance of all negotiations in a negotiation cluster (Line 1). Agent $a$ successively conducts each negotiation cluster based on the cluster sequence. During conducting a negotiation cluster (e.g., $Clu_j$), Agent $a$ concurrently conducts all involved negotiations in Cluster $Clu_j$ (Line 3), and Agent $a$ adopts the proposed strategy in Section 5.2 to negotiate with its opponents in all negotiations in Cluster $Clu_j$ (Lines 5-8). For each completed negotiation, Agent $a$ records the result of the negotiation and calculates the utility achieved from the negotiation (Lines 9-10). Agent $a$ concurrently conducts all negotiations in each negotiation cluster, and conducts
Algorithm 9 Clustered Negotiation Procedure of MCN

**Input:** a negotiation network $\Theta$, which involves MCN.

**Output:** the outcomes of all negotiations in the MCN.

1. **Pre-calculations:** (1) calculate the negotiation strength between every two negotiations in the MCN, (2) use a clustering algorithm to partition all $m$ negotiations into $k$ negotiation clusters, and (3) sort all negotiation clusters based on the sum of importance of all negotiations in each negotiation cluster to get a cluster sequence, e.g., $Clu_0 \rightarrow \cdots \rightarrow Clu_j \rightarrow \cdots \rightarrow Clu_{k-1}$, where $\sum_{N_i \in Clu_0} \nu_i \geq \cdots \geq \sum_{N_i \in Clu_j} \nu_i \geq \cdots \geq \sum_{N_i \in Clu_{k-1}} \nu_i$, and $\nu_i$ is the importance of Negotiation $N_i$.

2. **for** $j \leftarrow 0$ to $k - 1$ **do**

3. **while** not all negotiations in Cluster $Clu_j$ are completed **do**

4. **if** a counter-offer from its opponent in Negotiation $N_i$ in Cluster $Clu_j$ is received **then**

5. retrieve all the last values of issues in other negotiations which have interdependencies with issues in Negotiation $N_i$, and store the values in a list $lis$

6. **end if**;

7. adopt the proposed strategy in Section 5.2 while considering the values in the list $lis$

8. **if** Negotiation $N_i$ in Cluster $Clu_j$ is completed **then**

9. record the result of Negotiation $N_0$, i.e., success or failure, and calculate the utility achieved from Negotiation $N_i$, i.e., $U(N_i)$

10. **end if**;

11. **end while**;

12. $j \leftarrow j + 1$

13. **end for**;

14. **return** the outcomes of all negotiations.

all negotiation clusters successively until all $m$ negotiations are completed.

5.4 Experiment

In the experiment, an agent society is simulated with various MCN scenarios, and the performances of the three proposed MCN procedures are compared regarding negotiation efficiency and effectiveness.

In terms of experimental settings, detailed settings for every single negotiation and multiple negotiations are given, respectively. As this thesis does not focus on the single negotiation level, a widely used concession strategy, an issue procedure and a negotiation protocol for agents are adopted in every single negotiation with the setting of a series of random parameters. In the experimental settings for multiple negotiations, different MCN cases with the setting of various numbers of negotiations and issue interdependencies across MCN are presented to show the performance of the three proposed MCN procedures.
5.4.1 Experimental Settings for Single Negotiations

In the experiment, the following “time-dependent strategy” [FSJ98] is employed as the concession strategy for agents.

\[ U_a(t) = 1 - (1 - r_a)\left(\frac{t}{T_a}\right)^{\beta_a} \]  

(5.2)

where \( U_a(t) \in [0, 1] \) denotes the utility which Agent \( a \) achieves at negotiation period \( t \), \( r_a \in [0, 1] \) is Agent \( a \)’s reserved utility, \( T_a \) is Agent \( a \)’s deadline and \( \beta_a \) indicates Agent \( a \)’s concession rate.

In this work, the “package deal procedure” [FWJ06a] is adopted as the procedure to process multiple issues in single negotiation, and the “alternating offer protocol” [Rub82] is utilised for agents in every single negotiation.

To get general results of the proposed approach, all relevant parameters in every single negotiation are randomly selected, and the details are shown in Table 5.1.

| importance of negotiations | random |
| number of issues | random in [3, 8] |
| issue preference | random |
| deadline | random in [10, 20] |
| reserved utility | random in [0, 3, 0.4] |
| concession rate | random in \{((0, 1), 1, (1, 3))\} |

In the experiment, the value of the overall utility achieved from MCN is normalised in the range \([0, 1]\), and the overall utility achieved by an agent engaged in MCN \( N \), represented by \( U(N) \), is calculated by Equation (5.3).

\[ U(N) = \frac{1}{|N|} \times \sum_{\forall N_i \in N} \left( U(N_i) \times V(N_i) \right) \]  

(5.3)

where \( U(N) \in [0, 1] \), \( i \geq 0 \), \( N \) is the negotiation set, \( V(N_i) \) is the success possibility of Negotiation \( N_i \), which is calculated by Equation (5.4), and \( U(N_i) \) is the utility achieved by an agent in Negotiation \( N_i \).

\[ V(N_i) = \begin{cases} 0 & \text{if Negotiation } N_i \text{ is failed,} \\ 1 & \text{others.} \end{cases} \]  

(5.4)

5.4.2 Experimental Settings for Multiple Negotiations

In the experimental settings for multiple negotiations, a number of MCN scenarios with various settings are taken. To get general experimental results, the three proposed MCN procedures are tested in two cases based on different numbers of
negotiations and issue interdependencies. The detailed experimental settings in the two cases are shown in Table 5.2 and Table 5.3, respectively.

**Table 5.2:** Experimental settings for Case (a)

<table>
<thead>
<tr>
<th>number of negotiations</th>
<th>number of issue interdependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>50% number of negotiations</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.3:** Experimental settings for Case (b)

<table>
<thead>
<tr>
<th>number of negotiations</th>
<th>number of issue interdependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0% number of negotiations</td>
</tr>
<tr>
<td></td>
<td>30% number of negotiations</td>
</tr>
<tr>
<td></td>
<td>60% number of negotiations</td>
</tr>
<tr>
<td></td>
<td>80% number of negotiations</td>
</tr>
<tr>
<td></td>
<td>100% number of negotiations</td>
</tr>
</tbody>
</table>

Let us recall the mathematical representation of an issues interdependency described in Section 5.1, which is shown by Equation (5.5) as follows.

\[ \sum_{m=l}^{n} (\mu_m \times a_{N_j}^{t_k}(s_m)) \leq Q, \]

where \( \mu_m \) is the coefficient for the sub-offer on Issue \( s_m \) in Negotiation \( N_j \), \( Q \) is a constant, \( t_k \) indicates a negotiation period, and \( n-l+1 \) is the total number of issues involved in the issue interdependency.

The experimental settings for parameters in the mathematical representation of the issue interdependency are shown in Table 5.4.

**Table 5.4:** Parameters in issue interdependencies

<table>
<thead>
<tr>
<th>( \mu_m )</th>
<th>random in ([1, 3])</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>random in ([1, 5])</td>
</tr>
<tr>
<td>( n-l+1 )</td>
<td>random in ([2, 5])</td>
</tr>
</tbody>
</table>

In the experiment, every single MCN scenario is simulated 100 times by applying the three proposed MCN procedures, respectively. Specifically, the well-known “k-means” algorithm is employed as a clustering algorithm in the clustered negotiation procedure. Here, as this work does not focus on the optimization of MCN procedures, a series of values of \( k \) are selected in the “k-means” algorithm, and the average is taken as the experimental result for the clustered negotiation procedure.
5.4.3 Experimental Results

In the experiment, the performance of three MCN procedures regarding negotiation effectiveness and efficiency are tested, where the agent’s overall utility, the success rate of negotiations, and the number of negotiation rounds are reported. Agent’s overall utility indicates the overall utility achieved by an agent from MCN, which is calculated by Equation (5.3). The success rate of negotiations indicates the percentage of the number of successful negotiations in MCN. The number of negotiation rounds indicates the total number of negotiation rounds of conducting MCN.

The performances of three MCN procedures are tested in two cases (refer to Table 5.2 and Table 5.3), where

Case (a): the number of negotiations is 4, 8, 16, 30, respectively, and the number of issue interdependencies is set as 50% the number of negotiations, and
Case (b): the number of negotiations is set as 20, and the number of issue interdependencies is set as 0% the number of negotiations, 30% the number of negotiations, 60% the number of negotiations, 80% the number of negotiations, and 100% the number of negotiations, respectively.

1. Agent’s overall utility

Figure 5.3 shows agent’s overall utility achieved by conducting three negotiation procedures for MCN in Case (a). In Figure 5.3, the x-axis indicates the number of negotiations, and the y-axis indicates the overall utility achieved by the agent from the MCN. From Figure 5.3, it can be seen that, regardless of the number of negotia-
tions, the concurrent negotiation procedure could achieve the highest overall utility while the successive negotiation procedure might achieve the lowest overall utility.

Figure 5.4 shows agent’s overall utility achieved by conducting three negotiation procedures for MCN in Case (b). In Figure 5.4, the x-axis indicates the number of issue interdependencies (e.g., “30%” indicates that the number of issue interdependencies is 30% the number of negotiations), and the y-axis indicates the overall utility achieved by the agent from the MCN. From Figure 5.4, it can be seen that the agent’s overall utility achieved from the MCN goes down with the increase of the number of issue interdependencies. Moreover, regardless of the number of issue interdependencies, the concurrent negotiation procedure could achieve the highest overall utility while the successive negotiation procedure could achieve lowest overall utility.

2. Success rate of negotiations

Figure 5.5 shows the success rate of negotiations achieved by conducting three MCN procedures in Case (a). In Figure 5.5, the x-axis indicates the number of negotiations, and the y-axis indicates the success rate of negotiations achieved from the MCN. Figure 5.5 shows that, regardless of the number of negotiations, the concurrent negotiation procedure might achieve the best performance in the success rate of negotiations while the successive negotiation procedure could be the worst one.
Figure 5.5: Success rate of negotiations in Case (a)

Figure 5.6: Success rate of negotiations in Case (b)

Figure 5.6 shows the success rate of negotiations achieved by conducting three negotiation procedures for MCN in Case (b). In Figure 5.6, the x-axis indicates the number of issue interdependencies, and the y-axis indicates the success rate of negotiations achieved from the MCN. Figure 5.6 shows that the success rate of negotiations goes down with the increase of the number of issue interdependencies. This is because more issue interdependencies would make fewer negotiations successful.
Moreover, regardless of the number of issue interdependencies, the concurrent negotiation procedure could achieve the highest success rate of negotiations while the successive negotiation procedure might achieve the lowest success rate of negotiations.

3. Number of negotiation rounds

![Number of negotiation rounds in Case (a)](image)

**Figure 5.7**: Number of negotiation rounds in Case (a)

Figure 5.7 shows the total number of negotiation rounds when conducting MCN by employing three MCN procedures in Case (a). In Figure 5.7, the x-axis indicates the number of negotiations, and the y-axis indicates the total number of negotiation rounds when conducting the MCN. Figure 5.7 shows that, regardless of the number of negotiations, the successive negotiation procedure might achieve the best performance in negotiation efficiency while the concurrent negotiation procedure could be the worst one.

Figure 5.8 shows the total number of negotiation rounds of conducting the MCN by employing three MCN procedures in Case (b). In Figure 5.8, the x-axis indicates the number of issue interdependencies, and the y-axis indicates the total number of negotiation rounds when conducting the MCN. Figure 5.8 shows that the total number of negotiation rounds when conducting the MCN goes up with the increase of the number of issue interdependencies. This is because more issue interdependencies would make it more time-consuming in achieving agreements. Moreover, regardless of the number of issue interdependencies, the successive negotiation procedure could
Figure 5.8: Number of negotiation rounds in Case (b)

achieve the highest negotiation efficiency while the concurrent negotiation procedure might be the least efficient one.

5.5 Summary

This chapter firstly proposed a general MCN model, where multiple multi-issue negotiations were involved. Then, a negotiation strategy was proposed for agents to generate approximating Pareto-optimal offers. Three negotiation procedures were proposed for handling MCN, which were the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. Additionally, an experiment was carried out to show the different performances of the three proposed negotiation procedures. The experimental results showed that: (1) regarding negotiation effectiveness, the concurrent negotiation procedure could achieve the best performance, (2) regarding negotiation efficiency, the successive negotiation procedure could be the most efficient one, and (3) the clustered negotiation procedure could provide a well-balanced solution between negotiation effectiveness and efficiency. The proposed solution in this chapter could successfully fulfil Objective 3 of this thesis (refer to Section 1.5).
Chapter 6

A Trust-Based Approach for a Negotiation Society

To achieve Objective 4 of this thesis (refer to Section 1.5), this chapter proposes an approach for agents to handle MCN while considering social factors in an agent society. First, a general model for a negotiation society is proposed, and a hierarchical graph-based representation of a negotiation society is given, which includes an agent network layer and a negotiation network layer. Then, by considering issue interdependency across MCN and agents’ reputations, a trust-based approach is proposed for handling social factors of MCN in an agent society.

The outline of this chapter is as follows. Section 6.1 proposes a general model for a negotiation society. Section 6.2 proposes a trust-based model for a negotiation society. Section 6.3 presents a negotiation mechanism for an agent’s negotiation network. Section 6.4 demonstrates experimental results by using the proposed trust-based approach. Section 6.5 summarises this chapter.

6.1 A General Model for a Negotiation Society

This section proposes a general model for a negotiation society, which includes the formalisation of a negotiation society, and a hierarchical graph-based representation of a negotiation society.

6.1.1 Formalisation of a Negotiation Society

Definition 6.1.1. (Negotiation Society) A negotiation society indicates a society engaged by multiple agents conducting MCN with each other, and it is defined by a tuple $\Theta = < A, T, \Phi >$, where

- $\text{Set } A = \{a_0, \cdots a_i, \cdots a_k\} (i \geq 0)$ denotes a set of agents;
Set $T = \{T_{a_0}, \ldots, T_{a_k}, \ldots, T_{a_m}\}(i \geq 0)$ denotes a set of trust value sets for each agent, where $T_{a_i} = \{t_{a_i}|0 \leq j \leq k, j \neq i\}$ indicates a set of trust values for other agents rated by Agent $a_i$, i.e., Agent $a_i$ is the trustor and other agents are the trustees;

Set $\Phi = \bigcup_{i=0}^{k} \Psi(a_{i})$ denotes a set of agents’ negotiation networks, where $a_i \in A$, and $\Psi(a_i)$ indicates Agent $a_i$’s negotiation network (see Definition 6.1.2).

**Definition 6.1.2. (Agent’s Negotiation Network)** Agent $a_i$’s negotiation network indicates a network where Agent $a_i$ conducts MCN with other agents in a negotiation society, and it is defined as a set $\Psi(a_i) = \{N_{a_0}^a, A^a, S^a, V^a, F^a, T^a\}$, where

- $Set N_{a_0}^a = \{N_{a_0}^a, \ldots, N_{a_i}^a, \ldots, N_{a_m}^a\}(i \geq 0)$ denotes a negotiation set, where $N_{a_i}^a$ indicates a bilateral single-issue/multi-issue negotiation between Agent $a_i$ and its opponent;
- $Set A^a = \{a_i^{N_{a_0}^a}, \ldots, a_i^{N_{a_i}^a}, \ldots, a_i^{N_{a_m}^a}\}(i \geq 0)$ denotes a set of Agent $a_i$’s negotiation opponents, where for $\forall a_i^{N_{a_i}^a} \in A^a$ and $\forall N_{a_i}^a \in N_{a_0}^a$, $a_i^{N_{a_i}^a}$ is Agent $a_i$’s opponent in Negotiation $N_{a_i}^a$;
- $Set S^a = \{S_{N_{a_0}^a}, \ldots, S_{N_{a_i}^a}, \ldots, S_{N_{a_m}^a}\}(i \geq 0)$ denotes a set of issue sets for each negotiation, where $S_{N_{a_i}^a} = \{s_0, \ldots, s_j, \ldots, s_n\}(j \geq 0)$ denotes a set of issues in Negotiation $N_{a_i}^a$, where $s_j$ indicates a negotiation issue;
- $Set V^a = \{\nu_{N_{a_0}^a}^a, \ldots, \nu_{N_{a_i}^a}^a, \ldots, \nu_{N_{a_m}^a}^a\}(i \geq 0)$ denotes a set of importance of negotiations for Agent $a_i$, where $\nu_{N_{a_i}^a}^a \in (0, 1]$ indicates Agent $a_i$’s preference for the importance of Negotiation $N_{a_i}^a$, and $\sum_{i=0}^{m} \nu_{N_{a_i}^a}^a = 1$.
- $Set F^a = \{F_{N_{a_0}^a}, \ldots, F_{N_{a_i}^a}, \ldots, F_{N_{a_m}^a}\}(i \geq 0)$ denotes a set of offer sets for all negotiations, where $F_{N_{a_i}^a} = \{O_{a_i}, O_{a_i^{N_{a_i}^a}}\}$ indicates a set of offer sets for Negotiation $N_{a_i}^a$, where $O_{a_i} = \{o_{a_i}^{0}, \ldots, o_{a_i}^{l_{a_i^0}}, \ldots, o_{a_i}^{l_{a_i^l}}\}$ is a set of offers proposed by Agent $a_i$ in Negotiation $N_{a_i}^a$, and $O_{a_i^{N_{a_i}^a}} = \{o_{a_i}^{0_{a_i^0}}, \ldots, o_{a_i}^{l_{a_i^0}}, \ldots, o_{a_i}^{l_{a_i^l}}\}$ is a set of offers proposed by Agent $a_i$’s opponent $a_i^{N_{a_i}^a}$ in Negotiation $N_{a_i}^a$ (i.e., $o_{a_i}^{l_{a_i^l}}$ is the offer proposed by Opponent $a_i^{N_{a_i}^a}$ at negotiation period $t_r$);
- $Set I^a = \{i^0, \ldots, i^p, \ldots, i^q\}$, where $i^p = \{s_l \sim \ldots \sim s_m \sim \ldots \sim s_n|s_l, \ldots, s_m, \ldots, s_n \in S_{N_{a_i}^a} \times \ldots \times S_{N_{a_i}^a} \times \ldots \times S_{N_{a_i}^a}\}$ indicates a set of issue interdependencies (refer to Definition 5.1.1 in Chapter 5).
6.1.2 A Hierarchical Graph-Based Representation of a Negotiation Society

A negotiation society $\Theta = \langle A, T, \Phi \rangle$ is represented by a hierarchical graph $G = \{G_\text{A}, G_\text{N}\}$, where $G_\text{A}$ indicates an agent network layer and $G_\text{N}$ indicates a negotiation network layer, and each layer is represented by a graph (see Figure 6.1).

The agent network layer is represented by an undirected graph $G_\text{A} = \langle V_\text{A}, E_\text{A} \rangle$, where $V_\text{A} = A$ and $E_\text{A} = \{e_0, \ldots, e_i, \ldots, e_m\}$. Each node indicates an agent, and an edge is represented by $e_{ij} = \{(a_i, a_j)|a_i, a_j \in A\}$, where each edge $e_{ij}$ indicates a negotiation conducted between Agents $a_i$ and $a_j$.

The negotiation network layer is represented by a set of undirected graphs $G_\text{N} = \{G_{a_0}, \ldots, G_{a_i}, \ldots, G_{a_k}\}$, where $G_{a_i} = \langle V_{a_i}, E_{a_i} \rangle$ represents Agent $a_i$’s negotiation network, where $V_{a_i} = \{N_{a_i}, S_{a_i}\}$ indicates a set of negotiations $N_{a_i}$ with a set of negotiation issues $S_{a_i}$ engaged by Agent $a_i$, and $E_{a_i}$ indicates issue interdependencies across the MCN engaged by Agent $a_i$.

![Figure 6.1: An example of a negotiation society’s hierarchical graph-based representation](image)

Let us consider an example of a negotiation society $\Theta = \langle A, T, \Phi \rangle$, where $A = \{a_0, a_1, a_2, a_3\}$, $T = \{T_{a_0}, T_{a_1}, T_{a_2}, T_{a_3}\}$, $\Phi = \bigcup_{i=0}^{3} \Psi(a_i)$, where $\Psi(a_i) = \{N_{a_i}, A_{a_i}, S_{a_i}, V_{a_i}, F_{a_i}, I_{a_i}\}$ indicates Agent $a_i$’s negotiation network.
Figure 6.1 shows the hierarchical graph-based representation of the negotiation society $\Theta$. In the agent network layer, the agent set is $A = \{a_0, a_1, a_2, a_3\}$. In the negotiation network layer, there are totally four negotiations performed between four agents in a negotiation society, and each Agent $a_i$ conducts its own negotiation network $\Psi(a_i)$. For instance, in Agent $a_2$’s negotiation network, Agent $a_2$ conducts Negotiations $N_0$, $N_1$ and $N_3$ with Agents $a_0$, $a_1$ and $a_3$, respectively. In each agent’s negotiation network, issue interdependency across MCN might exist, where an edge between issues in a negotiation network layer indicates an issue interdependency and no edge indicates no issue interdependency. For instance, in Agent $a_2$’s negotiation network $\Psi(a_2)$, Agent $a_2$ conducts Negotiations $N_0$, $N_1$, $N_3$ with Agents $a_0$, $a_1$, $a_3$, respectively. Among these three negotiations, there are two issue interdependencies, which are $s_{00} \sim s_{11}$ and $s_{12} \sim s_{31}$.

### 6.2 A Trust-Based Model for a Negotiation Society

In the proposed trust-based model for a negotiation society, the following aspects are considered to calculate the value of an agent’s overall trust.

- **Directed Trust**: a directed trust value on Agent $b$ given by Agent $a$ can be represented by $DT_{ab} \in [0, 1]$, which depicts that Agent $a$ rates Agent $b$ by utilising its own experiences in an interaction with Agent $b$ to determine Agent $b$’s trustworthiness.

- **Undirected Trust**: an undirected trust value on Agent $b$ given by Agent $a$ can be represented by $UT_{ab} \in [0, 1]$, which describes Agent $a$ rates Agent $b$ by combining directed trust values on all other agents rated by Agent $a$ and directed trust values on Agent $b$ rated by all other agents to derive Agent $b$’s trustworthiness. Assuming the agent set is $A = \{a_0, \ldots, a_i, \ldots, a_k\}$, an undirected trust value $UT_{a_ia_j}$ can be calculated by Equation (6.1).

$$UT_{a_ia_j} = \frac{1}{k} \sum_{l=0, l \neq i}^{k} (DT_{a_ia_l} \times DT_{a_la_j}) \quad (6.1)$$

- **Self Confidence**: a self confidence value of Agent $a$ can be represented by $SC_a \in [0, 1]$, which indicates how much Agent $a$ believes itself.

- **Public Confidence**: a public confidence value of Agent $a$ can be represented by $PC_a \in [0, 1]$, which indicates how much Agent $a$ believes the public (i.e., all other agents in the negotiation society). The sum of a self confidence value
and a public confidence value of an agent (e.g., Agent \(a\)) should be 1, i.e., \(SC_a + PC_a = 1\).

- **Overall Trust**: an overall trust value on Agent \(b\) assessed by Agent \(a\) can be represented by \(OT_{ab} \in [0, 1]\), which depicts Agent \(b\)’s overall trustworthiness derived by Agent \(a\) by considering all above aspects, i.e., the directed trust, the undirected trust, the self confidence, and the public confidence. The value of an overall trust value \(OT_{ab}\) can be calculated by Equation (6.2).

\[
OT_{ab} = SC_a \times DT_{ab} + PC_a \times UT_{ab}
\]  

(6.2)

In order to avoid the situation, where new agents (i.e., agents have not conducted too many negotiations before) have fewer chances to be selected, Equation (6.3) is employed in the directed trust value calculation.

\[
f(x) = 1 - e^{-\frac{x}{\varphi}},
\]  

(6.3)

where \(\varphi\) is a coefficient to adjust the rate, and it could vary in different application domains.

In a negotiation society, a directed trust value \(DT_{ab}\) is based on Agent \(b\)’s performance of committing agreements in all historical negotiations between Agents \(a\) and \(b\). Agent \(a\) rates Agent \(b\) based on Agent \(b\)’s performance each time. The rating is called *satisfaction degree* in this work, and it can be represented by \(SD_{ab}^i\), which indicates Agent \(a\)’s rating on Agent \(b\) based on Agent \(b\)’s \(i\)th performance. In this work, the values of the satisfaction degree are in the range \([0, 1]\), where 0 is for completely unsatisfactory, 1 is for totally satisfactory, and 0.5 is for neutral. Each time after Agent \(a\) rates Agent \(b\), Agent \(a\) will store the satisfaction degree value in its local database and honestly broadcast the updated directed trust value on Agent \(b\) to all other agents in the negotiation society. Based on Agent \(a\)’s historical satisfaction degree values on Agent \(b\), the value of directed trust \(DT_{ab}\) can be calculated by Equation (6.4).

\[
DT_{ab} = \mu(SD_{ab}) \times \frac{1 - \sigma(SD_{ab})}{1 - e^{-\frac{1}{\varphi}}} \times (1 - e^{-\frac{\mu(SD_{ab})}{\varphi}}),
\]  

(6.4)

\[
\mu(SD_{ab}) = \frac{1}{n} \sum_{i=0}^{n} SD_{ab}^i,
\]  

(6.5)

\[
\sigma(SD_{ab}) = \sqrt{\frac{\sum_{i=0}^{n} (SD_{ab}^i - \mu(SD_{ab}))^2}{n}},
\]  

(6.6)
where $\mu(SD_{ab})$ indicates the mean of all historical satisfaction degree values $SD_{ab}$, which can be calculated by Equation (6.5), and $\sigma(SD_{ab})$ indicates the standard deviation of all historical satisfaction degree values $SD_{ab}$, which can be calculated by Equation (6.6).

In a negotiation society, agents would more likely negotiate with the agents who have good performance, i.e., high satisfaction degree values. Therefore, in Equation (6.4), an agent who has satisfaction degree values with larger mean and smaller standard deviation would be rated for a higher directed trust value.

### 6.3 A Negotiation Mechanism for an Agent’s Negotiation Network

This section proposes an agent’s offer generation strategy for conducting MCN in a negotiation society and a negotiation protocol for conducting MCN in an agent’s negotiation network.

#### 6.3.1 An Offer Generation Strategy for Agents in a Negotiation Society

In this section, a negotiation strategy for generating an approximating Pareto-optimal offers in an agent’s negotiation network is proposed, where agents’ trust values and issue interdependencies across MCN are considered.

1. **An offer generation strategy affected by opponents’ trust values**

In this work, the following time-dependent strategy [FSJ98] is extended as the concession strategy in every single negotiation for agents in handling MCN in a negotiation society.

$$U_a(t) = 1 - (1 - r_a)\left(\frac{t}{T_a}\right)^{\frac{1}{\beta_a}}$$  \hspace{1cm} (6.7)

where $U_a(t) \in [0, 1]$ denotes the utility which Agent $a$ achieves at negotiation period $t$, $r_a \in [0, 1]$ is Agent $a$’s reservation utility, $T_a$ is Agent $a$’s deadline and $\beta_a$ indicates Agent $a$’s concession rate.

In a negotiation society, agents with high overall trust values would probably gain more utility through highly satisfactory performances of committing agreements. Therefore, all agents in the negotiation society have to perform well after an agreement is made each time, which would make the negotiation society develop towards a positive way. Under this consideration, an agent’s strategy in each negotiation round would be affected by its opponents’ trust values, i.e., overall trust
values in this work. Equation (6.8) shows the proposed time-dependent strategy while considering trust values.

\[
U_a(t) = \begin{cases} 
1 - (1 - r_a)(\frac{t}{t_a})^{\frac{1}{\beta_a}} & \text{if } d_{ab}(t) \neq -1, \\
U_a(t - 1) & \text{if } d_{ab}(t) = -1
\end{cases}
\]

(6.8)

\[
d_{ab}(t) = OT_{ab}^t - OT_{ab}^{t-1}
\]

(6.9)

where in Equation (6.8), \(d_{ab}(t)\) indicates the difference between overall trust values for Agent \(b\) evaluated by Agent \(a\) at periods \(t\) and \(t - 1\), respectively, which can be calculated by Equation (6.9), and \(\beta_a\) is Agent \(a\)’s concession rate.

In this work, the concurrent negotiation procedure (refer to Section 5.3) is employed for agents to conduct MCN in a negotiation society, where agents concurrently conduct multiple negotiations in their own negotiation networks. In order to solve the problem of how to decide the order of conducting negotiations in each negotiation round, the following method of calculating an agent’s preference for conducting negotiations is proposed in this chapter.

A value of an agent’s preference (i.e., \(P_{ab}^{N_i}(t)\)) at period \(t\) for conducting Negotiation \(N_i\) with its opponent Agent \(b\) can be calculated by Equation (6.10).

\[
P_{ab}^{N_i}(t) = 1 - (1 - \nu_{N_i}) \times e^{-\frac{OT_{ab}^t}{\gamma}},
\]

(6.10)

where \(\nu_{N_i}\) is the importance of Negotiation \(N_i\) for Agent \(a\), \(OT_{ab}^t\) is Agent \(b\)’s overall trust value assessed by Agent \(a\) at period \(t\), and \(\gamma\) is a coefficient to control the decreasing speed of the overall trust value.

In Equation (6.10), \(\gamma\) is employed to balance the weight distribution between: (1) the importance of a negotiation for an agent, and (2) the overall trust value of the agent’s opponent assessed by the agent, i.e., agents have different beliefs on weight distributions between these two factors.

2. An offer generation strategy affected by both opponents’ trust values and issue interdependency across MCN

To simply describe the strategy, let us consider a negotiation scenario, where Agent \(a\) conducts MCN with its opponents. One of the negotiations in the MCN is Negotiation \(N_0\), Agent \(a\)’s opponent in Negotiation \(N_0\) is \(a_{N_0}\), negotiation issues are \(s_0\) and \(s_1\), and Issue \(s_1\) has an issue interdependency with an issue in another negotiation.

Figure 6.2 shows Agent \(a\)’s strategy affected by both opponents’ trust values and issue interdependency across MCN.
At period $t_0 - 1$ (see Figure 6.2), Agent $a$’s opponent $a_{N_0}$ proposes a counter-offer $o_{a_{N_0}}^{t_0 - 1}$ to Agent $a$. At period $t_0$, Agent $a$ firstly assesses the overall trust value for its opponent $a_{N_0}$, i.e., $OT_{a_{a_{N_0}}}$. During the process of Negotiation $N_0$, Opponent $a_{N_0}$’s overall trust value might be different as other agents in the negotiation society might update Opponent $a_{N_0}$’s trust values based on its performance in other negotiations. Assuming the satisfied area for Issue $s_1$ at period $t_0$ is $P^t_{a_{N_0}}$, Agent $a$’s strategies of proposing counter-offers are based on both Opponent $a_{N_0}$’s overall trust value and issue interdependency across MCN. At period $t_0$, if

1. **Issue $s_1$ is not affected by an issue interdependency involving Issue $s_1$ and the opponent’s overall trust value keeps same**
   Agent $a$ selects a counter-offer $o_{a_{N_0}}^a$, represented by a black square, which has the shortest distance with point $o_{a_{N_0}}^{t_0 - 1}$;

2. **Issue $s_1$ is affected by an issue interdependency involving Issue $s_1$ and the opponent’s overall trust value keeps same**
   Agent $a$ selects a counter-offer $\tilde{o}_{a_{N_0}}^a$, represented by a black dot, which simultaneously satisfies conditions of having the shortest distance with point $o_{a_{N_0}}^{t_0 - 1}$ and falling in the shaded area $P^t_{a_{N_0}}$, i.e., the shaded area $P^t_{a_{N_0}}$ is calculated by the mathematical restriction of an issue interdependency between Issue $s_1$ and an issue in another negotiations;

3. **Issue $s_1$ is affected by an issue interdependency involving Issue $s_1$ and the opponent’s overall trust value becomes higher**

---

**Figure 6.2:** Edgeworth-Bowley box for an agent’s strategy affected by opponent’s trust value and issue interdependencies
Agent $a$ selects a counter-offer $\hat{o}^{t_0}_a$ on the curve $\hat{C}^{t_0}_a$, represented by a black ellipse, which satisfies conditions of having the shortest distance with point $o^{t_0}_{aN_0}$ and falling in the shaded area $P^{t_0}_a$.

4. Issue $s_1$ is affected by an issue interdependency involving Issue $s_1$ and the opponent’s overall trust value becomes lower

Agent $a$ selects a counter-offer $\tilde{o}^{t_0}_a$ on the curve $\tilde{C}^{t_0}_a$, represented by a black ellipse, which satisfies conditions of having the shortest distance with point $o^{t_0-1}_{aN_0}$ and falling in the shaded area $P^{t_0}_a$.

By employing the above strategy for Agent $a$, negotiations are concurrently conducted until an offer is accepted or an deadline is reached.

6.3.2 A Negotiation Protocol for an Agent’s Negotiation Network

This section proposes a negotiation protocol for an agent to concurrently conduct MCN in its negotiation network. To achieve the concurrency in MCN, the transition system-based approach introduced in Chapter 5 is employed in the proposed negotiation protocol.

The protocol for an agent’s negotiation network is presented by Algorithm 10. In Algorithm 10, the input is Agent $a$’s negotiation network $\Psi(a)$ including a negotiation set $N^a$, a set of agent’s opponents $A^a$, a set of issue sets for each negotiation $S^a$, a set of importance of all negotiations $V^a$, a set of offer sets for all negotiations $F^a$, and an issue interdependency set $I^a$. The outputs of Algorithm 10 are the overall utility achieved by Agent $a$ engaged in the negotiation network and outcomes of all negotiations, i.e., success or failure. In Line 1, the algorithm calculates transition system-based representations for all involved negotiations $TS_{N_i}$. Then, Agent $a$ concurrently starts all negotiations involved in its negotiation network (Line 2). During the process of negotiations, if Agent $a$ receives latest trust values from other agents, the algorithm updates Agent $a$’s database where stores all other agents’ trust values (Lines 4-5). The algorithm calculates Agent $a$’s current negotiation preference to select a negotiation to conduct (e.g., Negotiation $N_l$) (Line 7). If Agent $a$ receives an offer from its opponent in a negotiation (e.g., $N_i$), Agent $a$ checks the issue interdependencies and retrieves all the last values of related issues in other negotiations. The algorithm adopts the proposed strategy for Agent $a$ in Section 6.3.1 to generate offers in negotiations with its opponents, and then updates the offer set $F^a$ (Lines 8-11). When a negotiation is completed, the algorithm records the outcome of the negotiation and calculates the utility achieved from the negotiation (Lines 13-14). All $m$ negotiations are concurrently conducted until all of them are completed. Al-
CHAPTER 6. A TRUST-BASED APPROACH FOR A NEGOTIATION SOCIETY

Algorithm 10 A negotiation protocol for MCN in an agent’s negotiation network

Input: Agent $a$’s negotiation network $\Psi(a) = \{N^a, A^a, S^a, V^a, F^a, I^a\}$, where $N^a = \{N^a_0, \ldots, N^a_i, \ldots, N^a_m\}$, $A^a = \{a^{N^a_0}, \ldots, a^{N^a_i}, \ldots, a^{N^a_m}\}$, $S^a = \{S^a_{N^a_0}, \ldots, S^a_{N^a_i}, \ldots, S^a_{N^a_m}\}$, $V^a \neq \emptyset$, $F^a = \emptyset$, and $I^a \neq \emptyset$

Output: overall utility achieved by Agent $a$ engaged in its negotiation network $\Psi(a)$ (i.e., $U(\Psi(a))$), and outcomes of all negotiations, i.e., success or failure

1: Pre-calculations: calculate the transition system-based representations of all negotiations in Set $N^a$.
2: start conducting all $m$ negotiations in the negotiation set $N^a$ concurrently.
3: while not all negotiations are completed do
4: if Agent $a$ receives latest trust values from other agents then
5: update Agent $a$’s local database where stores trust values for other agents
6: end if;
7: calculate the preference for conducting each negotiation by Equation (6.10) to select Negotiation $N_i \in N'$ with highest preference value and conduct Negotiation $N_i$
8: if agent receives an offer from its opponent in a negotiation, e.g., Negotiation $N_i$ then
9: (1) retrieve all the last values of issues in other negotiations which have interdependencies with issues in negotiation $N_i$, and store the values in a list $lis$,
10: (2) adopt the proposed strategy in Section 6.3.1 while considering values in the list $lis$, and
11: (3) update the offer set $F^a$
12: end if;
13: if a negotiation (e.g., $N_i$) is completed then
14: record the outcome of Negotiation $N_i$, i.e., success or failure, calculate utility achieved in Negotiation $N_i$, i.e., $U(N_i)$
15: end if;
16: end while;
17: calculate the overall utility achieved in the negotiation network $\Psi(a)$, i.e., $U(\Psi(a))$
18: return the overall utility $U(\Psi(a))$ and outcomes of all negotiations in $\Psi(a)$

Algorithm 10 calculates the overall utility achieved from the negotiation network, i.e., $(U(\Psi(a)))$ (Line 17). In the end, the algorithm returns the overall utility and the outcomes of all negotiations as the outputs (Line 18).

6.4 Experiment

In the experiment, the performance of the proposed approach with various settings of simulated negotiation societies is tested.

In terms of experimental settings, detailed settings for every single negotiation and negotiation societies are given, respectively. As this work does not focus on the single negotiation level, a widely used concession strategy, a negotiation procedure
and a negotiation protocol for agents are adopted in every single negotiation with the setting of a series of random parameters. In the experimental settings for negotiation societies, different cases of MCN scenarios with the setting of various numbers of negotiations and issue interdependencies across MCN are given.

6.4.1 Experimental Settings for Every Single Negotiation

In the experiment, the following “time-dependent strategy” [FSJ98] is employed as the concession strategy for agents in every single negotiation.

\[
U_a(t) = 1 - (1 - r_a)(\frac{t}{T_a})^{\beta_a}
\]  

(6.11)

where \(U_a(t) \in [0, 1]\) denotes the utility which Agent \(a\) achieves at negotiation period \(t\), \(r_a \in [0, 1]\) is Agent \(a\)’s reserved utility, \(T_a\) is Agent \(a\)’s deadline, and \(\beta_a\) indicates Agent \(a\)’s concession rate.

In this work, the “package deal procedure” [FWJ06a] is adopted as the negotiation procedure to process multiple issues in every single negotiation, and the “alternating offer protocol” [Rub82] is utilised for agents in every single negotiation.

To get general results of the proposed approach, all relevant parameters in every single negotiation are randomly selected, and the details are shown in Table 6.1.

**Table 6.1:** Parameters in every single negotiation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>importance of a negotiation</td>
<td>random</td>
</tr>
<tr>
<td>number of issues</td>
<td>random in ([3, 6])</td>
</tr>
<tr>
<td>issue preference</td>
<td>random</td>
</tr>
<tr>
<td>deadline</td>
<td>random in ([10, 20])</td>
</tr>
<tr>
<td>reserved utility</td>
<td>random in ([0, 3, 0.4])</td>
</tr>
<tr>
<td>concession rate</td>
<td>random in ({(0,1), 1, (1,3)})</td>
</tr>
</tbody>
</table>

6.4.2 Experimental Settings for Negotiation Societies

In the experimental settings for negotiation societies, there are totally 100 agents in a negotiation society, where each agent conducts MCN with other agents in the negotiation society. In order to get general experimental results, negotiation societies with various settings are tested, and the detailed experimental settings in different scenarios are shown in Table 6.2 and Table 6.3. For instance, in Case 4 of Scenario (a), the total number of negotiations involved in the negotiation society is random in the range \([45, 50]\), and the total number of issue interdependencies involved in the MCN conducted by the agent is 50% number of negotiations.
Table 6.2: Experimental settings for MCN in Scenario (a)

<table>
<thead>
<tr>
<th>cases</th>
<th>number of negotiations</th>
<th>number of issue interdependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>random in [4, 6]</td>
<td>50% number of negotiations</td>
</tr>
<tr>
<td>2</td>
<td>random in [8, 10]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>random in [16, 20]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>random in [45, 50]</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: Experimental settings for MCN in Scenario (b)

<table>
<thead>
<tr>
<th>cases</th>
<th>number of negotiations</th>
<th>number of issue interdependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>random in [45, 50]</td>
<td>0% number of negotiations</td>
</tr>
<tr>
<td>2</td>
<td>random in [45, 50]</td>
<td>20% number of negotiations</td>
</tr>
<tr>
<td>3</td>
<td>random in [45, 50]</td>
<td>50% number of negotiations</td>
</tr>
<tr>
<td>4</td>
<td>random in [45, 50]</td>
<td>80% number of negotiations</td>
</tr>
<tr>
<td>5</td>
<td>random in [45, 50]</td>
<td>100% number of negotiations</td>
</tr>
</tbody>
</table>

Let us recall a mathematical representation of the issues interdependency described in Section 5.1 of Chapter 5, which is shown as Equation (6.12).

\[
\sum_{m=l}^{n} \left( \mu_m \times o_{N_j}^t (s_m) \right) \leq Q,
\]

(6.12)

where \( \mu_m \) is the coefficient for the sub-offer on Issue \( s_m \) in Negotiation \( N_j \), \( Q \) is a constant, \( t_k \) indicates a negotiation period, and \( n-l+1 \) is the total number of issues involved in the issue interdependency.

The experimental settings for parameters in the mathematical representation of issue interdependencies are shown in Table 6.4.

Table 6.4: Parameters in issue interdependencies

<table>
<thead>
<tr>
<th>( \mu_m )</th>
<th>random in [1, 3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>random in [1, 5]</td>
</tr>
<tr>
<td>( n-l+1 )</td>
<td>random in [2, 5]</td>
</tr>
</tbody>
</table>

6.4.3 Experimental Results

In the experiment, the performance of the proposed approach regarding negotiation effectiveness and efficiency is tested, where the social welfare, the success rate of negotiations, and the number of negotiation rounds are reported. The social welfare indicates the overall utility achieved by all agents in a negotiation society. The success rate of negotiations indicates the percentage of the number of successful negotiations in a negotiation society. The number of negotiation rounds indicates the total number of negotiation rounds of conducting MCN in a negotiation society.

In order to show the performance of the proposed trust-based approach for
conducting MCN in a negotiation society, the following three scenarios regarding agents’ overall trust values are tested in the experiment.

1. **Overall Trust Stay**: agents’ overall trust values are not changed;

2. **Overall Trust Up**: agents’ overall trust values increase;

3. **Overall Trust Down**: agents’ overall trust values decrease.

In the experiment, every single scenario of negotiation societies is simulated 100 times. Based on the experimental settings, the experimental results, regarding the **social welfare**, the **success rate of negotiations**, and the **number of negotiation rounds** in Scenario (a) and Scenario (b), are reported as follows.

1. **Social welfare**

![Figure 6.3: Social welfare in Scenario (a)](image_url)

Figure 6.3 shows the average social welfare achieved by all agents in negotiation societies in Scenario (a). In Figure 6.3, the x-axis indicates the number of negotiations, and the y-axis indicates the average social welfare. From Figure 6.3, it can be seen that, with the increase of agents’ overall trust values, the social welfare achieved by all agents increases. Moreover, with the increase of the number of negotiations, the difference between social welfare achieved in the scenarios of “Overall Trust Down” and “Overall Trust Up” gets larger. It means that agents in a negotiation society have to increase their overall trust values to avoid achieving very low social welfare.
Figure 6.4 shows the average social welfare achieved by all agents in negotiation societies in Scenario (b). In Figure 6.4, the x-axis indicates the number of issue interdependencies (e.g., “30%” indicates that the number of issue interdependencies is 30% the number of negotiations), and the y-axis indicates the average social welfare. From Figure 6.4, it can be seen that, with the increase of agents’ overall trust values, the social welfare achieved by all agents increases. Moreover, with the increase of the number of issue interdependencies, the difference between social welfare achieved in the scenarios of “Overall Trust Down” and “Overall Trust Stay” gets smaller. It means when the number of issue interdependencies is small, agents’ overall trust values has a larger impact than issue interdependency on social welfare.

2. Success rate of negotiations

Figure 6.5 shows the average success rate of negotiations achieved from negotiation societies in Scenario (a). In Figure 6.5, the x-axis indicates the number of negotiations, and the y-axis indicates the average success rate of negotiations. Figure 6.5 shows that, regardless of the number of negotiations, the difference between success rates of negotiations in the scenarios of “Overall Trust Down” and “Overall Trust Up” is always large. It means the agents in a negotiation society should avoid decreasing their overall trust values to get a low success rate of negotiations.
Figure 6.5: Success rate of negotiations in Scenario (a)

Figure 6.6: Success rate of negotiations in Scenario (b)

Figure 6.6 shows the average success rate of negotiations achieved from negotiation societies in Scenario (b). In Figure 6.6, the x-axis indicates the number of issue interdependencies, and the y-axis indicates the average success rate of negotiations. Figure 6.6 shows that, with the increase of the number of issue interdependencies, the success rate of negotiations decreases. The difference between success rates of negotiations in the scenarios of “Overall Trust Down” and “Overall Trust Up” gets
smaller. It indicates that, with the increase of the number of issue interdependencies, issue interdependency has a larger impact than agents’ overall trust values on the success rate of negotiations.

3. The number of negotiation rounds

Figure 6.7: The number of negotiation rounds in Scenario (a)

Figure 6.7 shows the average total number of negotiation rounds when conducting MCN in negotiation societies in Scenario (a). In Figure 6.7, the x-axis indicates the number of negotiations, and the y-axis indicates the average total number of negotiation rounds when conducting MCN. Figure 6.7 shows that, when the overall trust values of agents’ opponents decrease, the number of negotiation rounds will increase. Moreover, with the increase of the number of negotiations, the difference between the number of negotiation rounds in the scenarios of “Overall Trust Down” and “Overall Trust Up” gets larger. It indicates that agents should avoid decreasing their overall trust values to achieve high time-efficiency, especially when the number of involved negotiations in a negotiation society is large.

Figure 6.8 shows the average total number of negotiation rounds of conducting MCN in negotiation societies in Scenario (b). In Figure 6.8, the x-axis indicates the number of issue interdependencies, and the y-axis indicates the average total number of negotiation rounds when conducting MCN. Figure 6.8 shows that, when
agents’ overall trust values decrease, the number of negotiation rounds will increase. With the increase of the number of issue interdependencies, the difference between the number of negotiation rounds in the scenarios of “Overall Trust Down” and “Overall Trust Up” gets slightly larger. It shows that regardless the number of issue interdependencies, agents in a negotiation society should increase their overall trust values to achieve high time-efficiency.

In summary, an agent’s overall trust value is an important factor to impact the outcome of conducting MCN in an agent society. From the above experimental results, it can be found that, regarding the social welfare, the success rate of negotiations, and the number of negotiation rounds, agents in a negotiation society should avoid decreasing their overall trust values to achieve high society welfare, success rate of negotiations and time-efficiency, which indicates that by employing the proposed trust-based approach, agents in a negotiation society have to perform well in committing agreements in negotiations.

6.5 Summary

This chapter proposed a trust-based approach for handling society factors of MCN in an agent society. A general model for a negotiation society was proposed, where multiple agents conducted their MCN with other agents in a negotiation society. In the proposed approach, by considering agents’ reputations and issue interdependency across MCN, an agent’s offer generation strategy was proposed. A negotiation
protocol for an agent’s negotiation network was also presented to concurrently handle MCN in a negotiation society. Experimental results showed that the proposed approach effectively and efficiently handled social factors of MCN in an agent society, so the proposed trust-based approach in this chapter could successfully fulfil **Objective 4** of this thesis (refer to Section 1.5).
Chapter 7

Conclusion and Future Work

The research of Multiple Concurrent Negotiations (MCN) is challenging and important in agent negotiation. However, MCN has not been paid much attention by the researchers in agent negotiation. In order to overcome the limitations of current work on MCN, this thesis focused on investigating the challenging research issues in MCN and developing solutions to these challenging research issues. In order to solve the challenging issues in MCN described in Section 1.4, four approaches were proposed in this thesis.

This chapter concludes the thesis and outlines the future work on the research of MCN.

7.1 Contributions of The Thesis

This thesis focused on the research of MCN, and the contributions of this thesis include follows.

1. An MCN model for concurrently handling multiple negotiations with issue interdependency

In the proposed MCN model, MCN was formally defined by employing a set-based theory and issue interdependency across MCN was mathematically represented by a directed graph in Chapter 3. According to relationships between negotiations in MCN, MCN was categorised into five types, which were the single MCN, the sequential MCN, the synchronized MCN, the merging MCN and the hybrid MCN. Three algorithms were proposed to convert an MCN’s graph-based representation to its Colored Petri Net (CPN)-based representation. By conducting the CPN-based representation of MCN, a CPN-based negotiation protocol was proposed to concurrently handle MCN. Through considering different negotiation scenarios in an experiment, the experimental results showed the effectiveness and efficiency of the proposed MCN model in handling MCN.
2. An MCN protocol for handling MCN in a dynamic changing negotiation environment

The proposed protocol considered a dynamic changing negotiation environment in MCN, where during the process of MCN, any new negotiation could be added into MCN and any ongoing negotiation could be removed from MCN. By extending the MCN model proposed in Chapter 3, an updated mechanism was newly introduced in the proposed MCN protocol for a dynamic changing negotiation environment, including graph-based representation updating, utility updating, and CPN-based representation updating. Through applying a serial of manipulations on the CPN-based representation of MCN, the changes of the negotiations in a dynamic changing negotiation environment were effectively and efficiently handled. By considering all possible positions of adding and removing negotiations, an experiment was carried out to show the effectiveness and efficiency of the proposed protocol in handling MCN in a dynamic changing negotiation environment.

3. Three negotiation procedures for handling MCN in different ways

Three negotiation procedures for MCN were proposed in this thesis, which were the concurrent negotiation procedure, the successive negotiation procedure, and the clustered negotiation procedure. The concurrent negotiation procedure processed negotiations concurrently. The successive negotiation procedure processed multiple negotiations one after another. The clustered negotiation procedure processed multiple negotiations in several steps: (1) all negotiations were firstly partitioned into several clusters, (2) negotiations in each cluster were processed concurrently, and (3) all clusters were processed one after another. An experiment was conducted to evaluate the performances of the three negotiation procedures. The experimental results showed that the three negotiation procedures performed differently in terms of negotiation effectiveness and efficiency. The successive negotiation procedure achieved the best performance in terms of negotiation efficiency. The concurrent negotiation procedure achieved the best performance in terms of negotiation effectiveness. The clustered negotiation procedure provided a well-balanced solution between negotiation effectiveness and efficiency.

4. A trust-based approach for handling social factors of MCN in an agent society

A general model for a negotiation society was proposed, where multiple agents conducted their MCN with other agents in an agent society. A hierarchical
CHAPTER 7. CONCLUSION AND FUTURE WORK

A graph-based representation of a negotiation society was also given, including an agent network layer and a negotiation network layer. In the proposed approach, the concurrency in MCN was handled by a transition system. In the proposed approach, by considering opponent’s reputation and issue interdependency across MCN, an agent’s offer generation strategy was proposed. A negotiation protocol for an agent in a negotiation society was also presented to concurrently handle MCN. Experimental results showed that the proposed trust-based approach could effectively and efficiently handle social factors of MCN in an agent society.

7.2 Future Work

Although the proposed solutions in this thesis have proved the effectiveness and efficiency of handling MCN by considering different aspects, there is still some room for MCN to improve in future work.

1. Appropriate MCN strategies

In this thesis, solutions in terms of MCN strategies were not studied. Well-known strategies for generating offers were employed in the proposed solutions for agents in their decision-making processes. During the process of MCN, each agent utilised the same strategy in the proposed approaches in this thesis. However, to optimise an agent’s negotiation outcome, each agent may need to have its own specific strategy to generate offers. Moreover, in a dynamic changing negotiation environment, agents may need to adaptively adjust their strategies to meet unpredictable changes in processing MCN. Therefore, the study of MCN strategies will be one of important tasks in future work.

2. Information privacy in MCN

In Chapter 6, a trust-based approach was proposed for handling social factors of MCN in an agent society. In the proposed approach, there was an assumption that all agents in agent society were honest in broadcasting trust values to other agents. Another future direction is to investigate information privacy problems of conducting MCN in an agent society, such as avoiding agents’ cheat in sharing information, revealing less information shared between agents in the society, etc.

3. MCN applications

This thesis focused on the theoretical study on addressing challenging research
issues in MCN. In the future, the study of MCN applications by employing the proposed solutions in real-world situations, such as the task allocation [CBH09, DVR+07], the supply chain [Sta08, Chr16], and the resource allocation [ALS11, GNT06], will be another research direction.
Bibliography


al Conference on Web Intelligence and Intelligent Agent Technology, volume 2, pages 399–406, Toronto, Canada, 2010. IEEE.


BIBLIOGRAPHY


