ABSTRACT

An agent may have to simultaneously perform more than one negotiations which may have different dependence relationships in the real world. Since the Concurrent Multiple Negotiation (CMN) mechanism is necessary for the agent to achieve agreements in multi-negotiation, it has become a very important research topic in multi-agent systems in recent years. However, how to build an appropriate model and how to design an effective CMN protocol become serious challenges in the area of negotiation. The motivation of this paper is to propose a model for defining CMN and a protocol for concurrently handling CMN. Firstly, related CMN definitions are presented. Secondly, a graph-based method for modeling CMN is proposed. Thirdly, a protocol based on Colored Petri Net technique for achieving concurrency of CMN is presented. The proposed approach provides a mechanism for an agent to achieve CMN. The experimental results show that proposed approach is able to handle CMN smoothly and efficiently through prediction and tradeoff.

Keywords
Multi-Agent, Concurrent Multiple Negotiation, Colored Petri Net

1. INTRODUCTION

Negotiation is a method for different agents to communicate with each other to achieve agreements for their goals [1, 2, 3]. In general, a negotiation focuses on a single negotiation matter, and the negotiation goal is to reach a mutual agreement between negotiators on this matter. Depending on the number of negotiation issues, agent negotiation can be classified into single issue negotiation and multi-issue negotiation [1]. According to the number of negotiators, negotiation can be categorised into bilateral negotiation and multilateral negotiation [4] where bilateral negotiation is performed between two negotiators and multilateral negotiation is accomplished by multiple negotiators. However, these two negotiation styles still focus on the single negotiation matter.

In recent years, Concurrent Multiple Negotiation (CMN) has attracted attention from researchers. By comparing with the single goal negotiation, CMN is trying to reach a series of agreements on concurrently processed negotiations under different goals, and these concurrent negotiations are somehow related with each other, and the outcomes will contribute together for the agent to achieve an overall negotiation goal.

In complicated negotiation environments, an agent may negotiate with different negotiators for different purposes in order to reach an overall goal [5]. For instance, a customer intends to loan from a bank and buy a house with the loan, thus, he/she needs to negotiate with a banker and a house seller concurrently. Whether the customer can afford a house depends on the result of negotiation with the banker. If the loan is lower than the house’s price, the customer has to buy a house with lower price. Likewise, if the loan is higher than the house’s price, the customer is able to buy a better house with a higher price. Therefore, these two negotiations are somehow related and they interact each other. The example above is common in many real world applications. Thus, how to solve the problems of CMN becomes an important research issue in the area of agent and multi-agent systems.

However, little work has been done on the research of CMN. The challenge for studying CMN is that there is no existing well-accepted model to define CMN and no existing benchmark to handle the concurrency in multiple negotiations. Most existing related work only partially considered the research problems in CMN. In 2010, Sim and Shi [6] proposed a concurrent negotiation mechanism for grid resource coallocation, which applied concurrent one-to-many negotiations to solve related problems. However, in their approach, the relationship among agents is simple by only considering a consumer and multiple resource providers. Khalid Mensour et al. [7] proposed an approach on setting that a buyer negotiates concurrently with multiple sellers over different objects. Because these multiple negotiations have no correlations to impact each other, these two approaches did not consider the relationship among multiple negotiations, and they cannot cover all the multiple negotiation scenarios in the real negotiation environments.

Although these state-of-the-art approaches present a number of solutions for multiple negotiations, most of them were proposed without considering the relationship among negotiations. Therefore, how to present the definitions of CMN and how to model CMN become a significant research issue.

The contributions of this paper include the followings: 1) the formal related definitions of CMN are given, including the CMN, dependence relationship, multi-negotiation round,
multi-negotiation goals; 2) a graph-based approach to represent CMN with the consideration of their bilateral relationships is proposed; and 3) an innovative negotiation protocol for CMN by employing the combination of graph-based presentation with a Colored Petri Net (CPN) approach is presented.

The remaining of this paper is organised as follows. Section 2 describes the problem formulation on CMN. Section 3 presents a graph-based modeling method for CMN. Section 4 proposes a CPN-based protocol for CMN. Section 5 gives the experimental results and analysis. Section 6 concludes the paper and outlines the future work.

2. PROBLEM FORMULATION

In this section, problem formulation of CMN is presented and the CMN is categorised into five types.

2.1 Definitions

In this subsection, CMN, dependence, multi-negotiation rounds and multi-negotiation goals are defined in our problem formulation.

Definition 1. (Concurrent Multiple Negotiation (CMN)) A CMN is represented by a set \( R = \{A_1, A_2, \ldots, A_i, \ldots, A_k\} (i \geq 1) \), where each negotiation \( A_i \) is a bilateral negotiation or a multilateral negotiation.

CMN describes a situation that an agent performs a number of different negotiations concurrently where each negotiation with different goals, different opponents, different issues and negotiations are somehow dependent on others. Formally, the dependence of negotiations is defined as follows.

Definition 2. (Dependence) For negotiations \( A_i, A_j \in R \), \( A_i \propto A_j \) indicates that negotiation \( A_j \) depends on negotiation \( A_i \), and \( A_i \) must start before negotiation \( A_j \) in each multiple negotiation round. The Dependence has two Properties and one Lemma as follows.

Property 1. (Unidirectionality) For \( \forall A_i, A_j \in R \), if it satisfies \( A_i \propto A_j \), it does not hold that \( A_j \propto A_i \), and vice versa.

This property indicates that negotiation \( A_i \) and \( A_j \) cannot depend on each other simultaneously.

Property 2. (Transitivity) For \( \forall A_i, A_j, A_k \in R \), it holds that:

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

This property indicates that the dependence between multiple negotiations can be transferred.

Lemma 1. For \( \forall A_i, A_j, A_k \in R \), it holds that:

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

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 \( A_i \propto A_k \) holds, it does not hold that \( A_k \propto A_i \), so Lemma 1 holds.

In order to distinguish different success degrees of a CMN \( R \) and analyse the protocol in a CMN, the definition of multi-negotiation round is presented as follows.

Definition 3. (Multi-Negotiation Round (MNR)) A MNR is represented by a set \( R_i, R_i \subseteq R, R_i \neq \emptyset, R_i = \{r_{i,1}, \ldots, r_{i,j}, \ldots, r_{i,k}\} \), where \( |R_i| = k_i \) and \( r_{i,j} \) indicates the \( j \)th negotiation (i.e., negotiation \( A_k \)) in a MNR \( R_i \), which satisfies \( r_{i,j} = A_k, f(r_{i,j}) = k_i, r_{i,1} \propto \cdots \propto r_{i,j} \propto \cdots \propto r_{i,k_i} \), and there is no such \( r_{i,k} \) that for \( \forall r_{i,k} \in R_i, r_{i,j} \propto r_{i,1} \) and \( r_{i,k} \propto r_{i,j} \) hold. If the number of MNRs in a CMN \( R \) is \( l \), then \( \bigcup_{i=1}^{l} R_i = R \).

In a MNR \( R_i \), if all involved negotiations \( r_{i,j} \in R_i \) reach a successful negotiation outcome, the MNR \( R_i \) is success. In a CMN, agent has its goal before performing CMN. Based on the goal, agent can make different decisions to achieve optimization. Based on whether each MNR \( R_i \) is success, Multi-Negotiation Goals are proposed as follows.

Definition 4. (Multi-Negotiation Goals (MNGs)) MNGs for a CMN are sorted into Complete Success Goal, Partial Success Goal and No Success Goal based on the number of successful \( R_i \) in a CMN.

- Complete Success Goal: The CMN achieves Complete Success Goal if all \( R_i \) in a CMN are success.
- Partial Success Goal: The CMN achieves Partial Success Goal if not all but at least one \( R_i \) in a CMN is success.
- No Success Goal: The CMN achieves No Success Goal if no \( R_i \) involved in a CMN is success.

2.2 The Classification of Concurrent Multiple Negotiation

In a CMN \( R \), according to the dependence between its MNRs, the CMN \( R \) can be categorised into five types, i.e., Single Negotiation CMN, Sequential CMN, Synchronized CMN, Merging CMN and Hybrid CMN.

1. Single Negotiation CMN: A Single Negotiation CMN \( N_{sng} \) indicates that a CMN includes only one MNR and the MNR includes only one negotiation, i.e., \( ( R_1 = N ) \land ( |R_1| = |N| = 1 ) \).

2. Sequential CMN: A Sequential CMN \( N_{seq} \) indicates that a CMN includes only one MNR and the MNR includes more than one negotiations, i.e., \( ( R_1 = N ) \land ( |R_1| = |N| > 1 ) \).

3. Synchronized CMN: A Synchronized CMN \( N_{sng} \) indicates that a CMN contains multiple MNRs and all MNRs start with the same negotiation, i.e.,

\[ \forall \forall R_i \in R, i > 1, \]

\[ r_{1,1} = \cdots = r_{1,j} = \cdots = r_{1,k} = \bigcap_{i=1}^{l} R_i \]

4. Merging CMN: A Merging CMN \( N_{mer} \) shows that a CMN contains multiple MNRs and all MNRs end with the same negotiation, i.e.,

\[ \forall \forall R_i \in R, i > 1, \]

\[ r_{1,k_i} = \cdots = r_{1,k} = \cdots = r_{1,k_i} = \bigcap_{i=1}^{l} R_i \]

5. Hybrid CMN: A Hybrid CMN \( N_{hyb} \) describes that a complex CMN is a combination of \( N_{seq}, N_{sng} \) and/or \( N_{mer} \), i.e.,

\[ \forall \forall R_i \in R, i > 1, \bigcap_{i=1}^{l} R_i \neq \emptyset \]

Based on the above definitions, it can be seen that a \( N_{sng} \) or a \( N_{mer} \) is a special \( N_{hyb} \) where it either starts or ends with the same negotiation.
3. A GRAPH-BASED METHOD FOR CMN MODELING

In order to model CMN, a method based on graph theory [8] is proposed, which contains representation, logic calculation and utility calculation of a CMN.

3.1 Representation of a CMN

Definition 5. (Graph Representation) A CMN N is represented by a directed graph \( G = (V,E) \), where \( V \) indicates vertexes in \( G \) \( (V = N) \), and \( E \) indicates dependences between negotiations in a CMN N. For any two negotiations \( A_i, A_j \), if \( A_i \sim A_j \), then \( e_{ij} = (A_i, A_j) \in E \).

Based on the classification of CMN, the graph representations of five types CMN are shown in Figure 1.

Figure 1: The graph representations of five types CMN

3.2 Logic Calculation of a CMN

In order to indicate the negotiation outcome, logic calculation of the negotiation \( A_i \), the MNR \( R_i \), and the CMN N are defined as follows.

Definition 6. (Logic of Negotiation) Let \( L(A_i) = \{True, False, Unsure\} \) be logic of negotiation \( A_i \), where \( L(A_i) = True \) indicates negotiation \( A_i \) is success, \( L(A_i) = False \) indicates negotiation \( A_i \) is failed and \( L(A_i) = Unsure \) indicates negotiation is still ongoing.

Definition 7. (Logic Conjunction) Let \( L(R_i) \) be the logic representation of MNR \( R_i \), where \( L(R_i) = \{\forall A_j \in R_i \wedge L(A_j)\} \). \( L(A_i) \wedge L(A_j) \) is logic representation of \( A_i \sim A_j \), where \( L(A_i) \wedge L(A_j) \) indicates conjunction logic relationship of \( A_i \sim A_j \). \( L(A_i) \wedge L(A_j) \) has the same truth table as \( L(A_i) \wedge L(A_j) \), but the symbol \( \wedge \) doesn’t satisfy associativity and commutativity.

Definition 8. (Logic Disjunction) Let \( L(N) = \bigvee_{R_i \in N} L(R_i) \) indicate the logic representation of a CMN N, where \( \bigvee \) indicates disjunction of MNR \( R_i \) in a CMN N.

Based on the logic description above, equations to calculate the success possibility of a CMN N are proposed as follows.

Let \( V(A_i) \) indicate the success possibility of the negotiation \( A_i \), and \( V(A_i) \) is defined in Equation (1) by considering its corresponding \( L(A_i) \) as follows.

\[
V(A_i) = \begin{cases} 
0 & \text{if } L(A_i) = False, \\
1 & \text{if } L(A_i) = True, \\
x \in (0,1) & \text{if } L(A_i) = Unsure.
\end{cases}
\]  \hspace{1cm} (1)

where \( x \) indicates the success possibility of the negotiation \( A_i \). Based on Equation (1), the success possibility of the MNR \( R_i \) and the CMN N can be calculated by Equation (2) and Equation (3) as follows.

\[
V(R_i) = \prod_{A_j \in R_i} V(A_j) \hspace{1cm} (2)
\]

\[
V(N) = \frac{\sum_{R_i \in N} V(R_i)}{|R_i|} \hspace{1cm} (3)
\]

Let \( \Omega_s \in [0,1] \) donate the value of multi-negotiation goal of a CMN defined in Definition 4 where \( \Omega_s = 1 \) indicates Complete Success Goal of a CMN, and \( \Omega_s = 0 \) indicates Partial Success Goal of a CMN and \( \Omega_s = 0 \) indicates the No Success Goal of a CMN. Therefore, the outcome of the multi-negotiation goal of a CMN N \( Out(\Omega_s) \) is defined as follows.

\[
Out(\Omega_s) = \begin{cases} 
success & \text{if } V(N) \geq \Omega_s, \\
failure & \text{if } V(N) < \Omega_s.
\end{cases}
\]  \hspace{1cm} (5)

3.3 Utility Calculation of a CMN

In this subsection, a method for utility calculation of a CMN is presented. The followings are the definitions of utility calculation of a MNR \( R_i \) and a CMN N.

\( U(R_i) \) be the utility of a negotiation \( A_i \), and based on the definition of \( V(A_i) \) and \( V(R_i) \), the utility of a MNR \( R_i \) and a CMN N are defined by the following two equations, respectively.

\[
U(R_i) = \sum_{A_j \in R_i} (\omega_j \times U(A_j) \times V(A_j)) \hspace{1cm} (6)
\]

\[
U(N) = \frac{\sum_{R_i \in N} U(R_i) \times V(R_i)}{|R_i|} \hspace{1cm} (7)
\]

where \( U(R_i), U(N) \in [0,1] \) and \( \omega = (\omega_1, \omega_2, \cdots, \omega_n) \), \( \sum_{i=1}^{n} \omega_i = 1 \). \( \omega \in [0,1] \) represents agent’s preference for negotiation \( A_i \) in a CMN N.

\[\begin{align*}
\text{(a) Single Negotiation CMN} \quad & \text{(b) Sequential CMN} \\
\text{(c) Synchronized CMN} \quad & \text{(d) Merging CMN} \\
\text{(e) An example of Hybrid CMN}
\end{align*}\]
Equation (5) presents a method to calculate the utility of the MNR $R_i$. The MNR presents sequential dependence between negotiations. In a MNR $R_i$, agent has different "preference" for different negotiations which have different utilities and different negotiation results. The MNR shows concurrent dependence between different negotiations in a CMN $N$ as well.

The following CMN $N = \{A_1, A_2, A_3, A_4\}$ is taken as an example to show how calculation method described above works.

![Diagram of a Synchronized CMN](Figure 2: An example of a Synchronized CMN)

In the example above, the MNRs are $N = \{R_1, R_2, R_3\}$ where $R_1 = \{A_1, A_2\}$, $R_2 = \{A_1, A_3\}$, $R_3 = \{A_1, A_4\}$. For computing simplistically, we set $\omega_1 = 0.6$, $\omega_2 = 0.2$, $\omega_3 = 0.1$, $V(A_1) = V(A_2) = V(A_3) = V(A_4) = 1$, $U(A_1) = 0.5$, $U(A_2) = 0.4$, $U(A_3) = 0.3$, $U(A_4) = 0.2$, then $U(R_1) = \omega_1 \times U(A_1) \times V(A_1) + \omega_2 \times U(A_2) \times V(A_2) = 0.38$, $U(R_2) = \omega_1 \times U(A_1) \times V(A_1) + \omega_3 \times U(A_3) \times V(A_3) = 0.33$, $U(R_3) = \omega_1 \times U(A_1) \times V(A_1) + \omega_4 \times U(A_4) \times V(A_4) = 0.32$.

and

$$U(N) = \sum_{i=1}^{3} \left( U(R_i) \times V(R_i) \right) / 3 = 0.34$$

In Figure 2, three MNRs have the same vertex $A_1$, and negotiation $A_2, A_3, A_4$ all depend on negotiation $A_1$. In other words, if negotiation $A_1$ fails, none of negotiations $A_2, A_3, A_4$ can achieve success. So, negotiation $A_1$ is the most important negotiation in $N = \{A_1, A_2, A_3, A_4\}$. Coincidentally, the methods for calculating the utility of a CMN $N$ in Equations (5) and (6) reflect the feature which shows the importance of negotiations in a CMN $N$.

4. A CPN-BASED PROTOCOL FOR CMN

In this section, a CPN-based protocol in CMN is proposed to show how agents operate in such scenarios concurrently. Furthermore, analysis of the proposed protocol which provides prediction to improve efficiency of CMNs is presented.

4.1 Connection between CPN and CMN

CPN is a useful tool to support concurrency of systems. By taking advantages of CPN, agents can flexibly control protocols for CMN.

A CPN is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ [9]. $P$ is the set of places, and $T$ indicates the set of transitions. $A$ is the set of arcs and $A \subseteq P \times T \cup T \times P$. $\Sigma$ is a finite non-empty colour set, and $V$ shows a set of variables. $C$ indicates the colour set function, which assigns a colour set to each places, and $C : P \rightarrow \Sigma$ and $C(p) \in \Sigma$. $G$ is a guard function $G$, where $Type[G(t)] =Bool$. $E$ and $I$ are arc expression function and initialisation function, respectively.

In our CPN-based CMN model, transitions represent negotiations, and places represent states of negotiations. The inputs and outputs of negotiations are shown by arc directions, and the token $(1A, m)$ indicates that one offer from the $m$th round of negotiation $A$ (i.e., enable transition $t_A$). At the beginning of running CPN-based protocol, an initial token exists in initial places of each MNRs, which is used to activate its MNR in the protocol. If the transition next to the initial place is activated by initial token, the protocol will go on based on the result of the negotiation. If the negotiation completes a round, the token will be moved to the next place(s) according to the arc direction(s). In order to handle the concurrency of negotiations, a backward arc is added to show that different negotiations can be performed concurrently.

Based on the categories of CMN described in previous section, detailed CPN descriptions of five types CMN are shown as follows.

1. Single Negotiation CMN

![Diagram of a Single Negotiation CMN](Figure 3: The CPN of a Single Negotiation CMN)

In the CPN of a Single Negotiation CMN, an initial token $(1A, 1)$ acts as an input to activate transition $t_{A_1}$. If transition $t_{A_1}$ is enabled in the $m$th round, it outputs $(1A, m + 1)$ to place $P_2$ and $P_1$, respectively, where token $(1A, m + 1)$ in place $P_2$ acts as a finished mark of Single Negotiation CMN and the token $(1A, m + 1)$ makes transition $t_{A_1}$ enabled for the next new round.

2. Sequential CMN

![Diagram of a Sequential CMN](Figure 4: The CPN of a Sequential CMN)

In the CPN of a Sequential CMN, there is only one MNR in a CMN, an initial token $(1A_1, 1)$ acts as an input to activate transition $t_{A_1}$. If transition $t_{A_1}$ is enabled in the $m$th round, it outputs $(1A_2, m)$ and $(1A_1, m + 1)$ to place $P_2$ and $P_1$, respectively, where token $(1A_2, m)$ in place $P_2$ goes on activating transition $t_{A_2}$ and token $(1A_1, m + 1)$ makes transition $t_{A_1}$ enabled for the next new round. The running and activation of posterior transitions are the same.

3. Synchronized CMN
In the CPN of a Hybrid CMN, the running of each MNR can be viewed as several Sequential CMNs. For the first MNR $R_1 = \{A_1, A_2, A_3\}$, an initial token $(1A_1, 1)$ acts as an input to activate transition $t_{A_1}$. If transition $t_{A_1}$ is enabled in the $m$th round, it outputs $(1A_2, m)$ and $(1A_1, m + 1)$ to place $P_2$ and $P_1$, respectively. If transition $t_{A_2}$ is enabled, it outputs $(1A_3, m)$ to place $P_3$. The other three MNRs in the example of a Hybrid CMN above perform as the same as the first MNR $R_1$.

4.2 CPN-Based Modeling of Concurrent Multiple Negotiation

In this subsection, algorithms for converting a directed graph $G$ to its corresponding CPN $C$ are presented. Algorithm 1 shows how to convert a single negotiation $A_i$ to its CPN. Algorithm 2 presents the conversion of a multiple negotiation round $R$. Algorithm 3 describes a method to convert a concurrent multiple negotiation $N$ to its CPN.

**Algorithm 1** The Conversion from a Graph to a CPN for a single negotiation $A_i$ ($Conv(A_i)$)

Input: a single negotiation $A_i$.
Output: a corresponding CPN $C_{A_i} = (P, T, A, \Sigma, V, C, G, E, I)$ based on the input single negotiation $A_i$

1: \[ P \leftarrow \{P_1, P_2\} \]
2: \[ T \leftarrow \{t_{A_i}\} \]
3: \[ A \leftarrow \{(P_1, t_{A_i}), (t_{A_i}, P_2), (t_{A_i}, P_1)\} \]
4: \[ \Sigma \leftarrow \{(1A_i, 1), (1A_i, 2), \ldots (1A_i, m), \ldots (1A_i, n)\} \]
5: \[ V \leftarrow \{m, A_i\} \]
6: \[ C(P_1) \leftarrow \emptyset \]
7: \[ C(P_2) \leftarrow \emptyset \]
8: for \( j \leftarrow 1 \text{ to } m \) do
9: \[ C(P_1) \leftarrow C(P_1) \cup (1A_i, j) \]
10: \[ C(P_2) \leftarrow C(P_2) \cup (1A_i, j) \]
11: end for;
12: \[ C(P_1) \leftarrow C(P_1) \cup (1A_i, m + 1) \]
13: \[ C \leftarrow \{C(P_1), C(P_2)\} \]
14: \[ E \leftarrow \{t_{A_i} \rightarrow \{1A_i, m\}\} \]
15: \[ I(P_1) \leftarrow \{1A_i, m\} \]
16: \[ I(P_2) \leftarrow \emptyset \]
17: \[ I \leftarrow \{I(P_1), I(P_2)\} \]
18: if negotiation $A_i$ is success then
19: \[ G(t_{A_i}) \leftarrow \text{true} \]
20: else if negotiation $A_i$ is failed then
21: \[ G(t_{A_i}) \leftarrow \text{false} \]
22: end if;
23: \[ G \leftarrow G(t_{A_i}) \]
24: \[ M_0(P_1) \leftarrow (1A_i, m) \]
25: \[ M_0(P_2) \leftarrow \emptyset \]
26: \[ M_0(P_1) \leftarrow \{M_0(P_1), M_0(P_2)\} \]
27: return $C_{A_i}$.

In Algorithm 1, the input is a single negotiation $A_i$, and the output is its corresponding CPN $C_{A_i}$. Firstly, it creates the place set $P$, the transition set $T$, the arc set $A$, the colour set $\Sigma$ and the variable set $V$ (Lines 1-5). It initializes colour set functions of place $P_1$ and $P_2$ as empty sets (Lines 6-7), then it carries on a loop to output colour set function set $C$ (Lines 8-13). It sets arc expression function $E$ (Line 14). Then it sets initialisation function set $I$ (Lines 15-17). Afterwards, It outputs guard function set $G$ by conditional
At first, it utilizes Algorithm 1 to convert the graph of each judgment (Lines 18-23). It sets markings of place P to its CPN. The input of the algorithm is a MNR \( R_i \), and its output is its corresponding CPN \( C_{R_i} \). Algorithm 2 converts the graph of each MNR to its CPN based on the input MNR \( R_i \). The input of the algorithm is a MNR \( R_i \), and its output is its corresponding CPN \( C_{R_i} \). Algorithm 3 converts the graph of a concurrent multiple negotiation \( N \) to its CPN. The output of the algorithm is a CPN \( C_N \). Algorithm 4 performs protocol analysis in a MNR \( R_i \). The output of the algorithm is a CPN \( C_{R_i} \).

**Algorithm 3 The Conversion from a Graph to a CPN for a concurrent multiple negotiation \( N \)**

**Input:** a CMN \( N = \{A_1, A_2, \ldots, A_n\} \) and its MNRs \( N = \{R_1, \ldots, R_i, \ldots, R_n\} \)

**Output:** a corresponding CPN \( C_N = (P, T, A, \Sigma, V, C, G, E, I) \) based on the input CMN \( N \)

1. for \( i \leftarrow 1 \) to \( n \) do
2. \( (P_i, T_i, A_i, \Sigma_i, V_i, C_i, G_i, E_i, I_i) \leftarrow Trans_2(R_i) \);
3. end for;
4. \( T \leftarrow \bigcup P \);
5. \( P \leftarrow \bigcup P_i \);
6. \( P \leftarrow P \cup P_{n+1} \);
7. for \( j \leftarrow 2 \) to \( n \) do
8. \( V \leftarrow V \cup V_i \);
9. \( \Sigma \leftarrow \Sigma \cup \Sigma_i \);
10. \( C \leftarrow C \cup C_i \);
11. \( G \leftarrow G \cup G_i \);
12. \( E \leftarrow E \cup E_i \);
13. \( I \leftarrow I \cup \{I_i\} \);
14. \( j + + ; \)
15. end for;
16. \( I \leftarrow \{I_1, I_2, \ldots, I_n\} \);
17. \( M_0(P_1) \leftarrow \{A_1, m\} \);
18. for \( i \leftarrow 1 \) to \( n \) do
19. \( M_0(P_{n+1}) \leftarrow \emptyset \);
20. end for;
21. return \( C_N \).

**Algorithm 4 CPN-Based Protocol Analysis in a MNR \( R_i \)**

**Input:** a CPN \( C_{R_i} = (P, T, A, \Sigma, V, C, G, E, I) \), calculation function \( V \) and utility function \( U \).

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

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

In Algorithm 4, it performs protocol analysis in a MNR \( R_i \). It inputs a CPN \( C_{R_i} \), calculation function \( V \) and utility function \( U \), then outputs \( V(R_i) \) and \( U(R_i) \). It sets a loop to make decisions whether a transition is enabled (Line 1). For each transition \( t_{k-1} \), if its result of guard function is false, then the agent terminates \( R_i \) at once and sets \( V(r_{k-1}) = 0 \), and if the result of guard function is true, then the agent sets \( V(r_{k-1}) = 1 \) (Lines 2-7). When the loop is completed, the agent computes \( V(R_i) \) and \( U(R_i) \) as the outputs (Lines 9-11).

### 4.3 CPN-Based Protocol Analysis

In this subsection, analysis of the protocol for a CMN is given. At first, Algorithm 4 presents protocol analysis in each MNR \( R_i \), and shows how agent executes CPN of each MNR \( R_i \) concurrently. Then, based on the outcomes of each MNR \( R_i \), Algorithm 5 proposes the protocol analysis in a CMN \( N \) and generates the negotiation results (success or failure) through comparing utility \( C_N \) and MNG \( \Omega_R \).
In Algorithm 5, it performs protocol analysis in a CMN

**Algorithm 5 CPN-Based Protocol Analysis in a CMN N**

**Input:** a CPN $CN = (P, T, A, \Sigma, V, C, G, E, I)$, the MNG $\Omega_a$, calculation function $V$ and utility function $U$.

**Output:** success or failure.

1: execute CPN of each MNR $R_i$ concurrently based on Algorithm 4;
2: while at least one MNR $R_i$ is finished do
3: calculate $V(N)$;
4: calculate $U(N)$;
5: if $U(N) < \Omega_a$ then
6: keep executing other unfinished MNRs;
7: else if $U(N) \geq \Omega_a$ then
8: terminate whole CPN and quit;
9: return success;
10: end if;
11: end while;
12: while whole CPN is completed do
13: calculate $V(N)$;
14: calculate $U(N)$;
15: if $U(N) \geq \Omega_a$ then
16: return success;
17: else if $U(N) < \Omega_a$ then
18: return failure;
19: end if;
20: end while;
21: return success or failure.

$N$ where the inputs are a CPN $CN$, the MNG $\Omega_a$, calculation function $V$, utility function $U$ and the output is the result of protocol analysis, which is success or failure. At first, it performs Algorithm 4 to execute CPN of each MNR $R_i$, concurrently (Line 1). Agent computes $V(N)$ and $U(N)$ as soon as one MNR $R_i$ has been completed (Lines 2-4). According to the results of $V(N)$ and $U(N)$, the agent can make different decisions. If the utility of a CMN $N$ is less than the MNG $\Omega_a$, the agent keeps executing CPN of other unfinished MNRs while if the utility of a CMN $N$ is no less than the MNG $\Omega_a$, the agent terminates whole CPN and quits at once (Lines 5-9). The algorithm shows that if a completed MNR $R_i$ achieves the MNG $\Omega_a$, then it outputs “success” for the whole CMN $N$. Because there is no need to execute other unfinished MNRs, it can improve efficiency in some extent. Then the agent keeps on calculating $V(N)$ and $U(N)$ until the whole CPN is completed. It compares the utility of a CMN $N$ with MNG $\Omega_a$ to make different decisions (Lines 12-20). In the end, it returns the result of protocol analysis.

In the algorithms above, an approach for an agent to analyse the protocol for its CMN is proposed. During running of the protocol, if it is found that it has already achieved the goal ($U(N) \geq \Omega_a$) even if not all MNRs have been completed, the agent will terminate the CPN of and quit from the process immediately. By using this way, the efficiency of CMN can be improved greatly when facing a large scale CMN.

5. EXPERIMENT

An experiment has been conducted to evaluate the performance of the proposed CPN-based protocol for CMN.

5.1 Experimental Setting

Suppose that Agent $a$ concurrently preforms five negotiations with five other agents in a CMN $N = \{A_1, A_2, A_3, A_4, A_5\}$. In the experiment, we take the following Hybrid CMN in Figure 8 as an example.

Figure 8: A Hybrid CMN

where $\omega_1 = \omega_2 = \omega_3 = \omega_4 = \omega_5 = 0.2$. The parameters for five negotiations in the CMN are listed in Table 1. For simplification, it contains one issue and agent $a$ has one opponent in each negotiation. The deadlines for each agent are selected randomly between 10 rounds and 20 rounds. In the CMN $N$, the value of multiple negotiation goal of agent $a$ is $\Omega_a = \frac{1}{2}$.

A single-issue negotiation model [11] is employed to process each single negotiation and every agent adopts random concession strategies (Conceder, Linear or Boulware) [11]. Utility function $U^a_\omega$ for each single negotiation is described as follows where $a$ indicates agent and $o$ indicates counter-offer.

$$U^a_\omega(o) = \text{Reserved Offer} - o$$

where 

$$\omega_1 = \omega_2 = \omega_3 = \omega_4 = \omega_5 = 0.2.$$  

5.2 Experimental Result

In this experiment, CPN Tools [12] was employed to handle concurrency in a CMN, which showed how multiple negotiations performed concurrently. Firstly, by using Algorithm Cover$_N(N)$, the CMN was converted from a graph to its CPN and then the CPN was shown in CPN Tools. Figure 9 shows that the CMN $N = \{A_1, A_2, A_3, A_4, A_5\}$ was executed by employing CPN Tools.

Table 1: Parameters for negotiations in a CMN $N$

<table>
<thead>
<tr>
<th>Negotiation</th>
<th>Agent $a$</th>
<th>[Initial Offer, Reserved Offer]</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>A$_1$</td>
<td>$a$</td>
<td>(300k, 500k)</td>
<td>17</td>
</tr>
<tr>
<td>A$_2$</td>
<td>$b_1$</td>
<td>(300k, 400k)</td>
<td>15</td>
</tr>
<tr>
<td>A$_3$</td>
<td>$b_2$</td>
<td>(400k, 500k)</td>
<td>12</td>
</tr>
<tr>
<td>A$_4$</td>
<td>$b_3$</td>
<td>(400k, 500k)</td>
<td>10</td>
</tr>
<tr>
<td>A$_5$</td>
<td>$b_4$</td>
<td>(400k, 500k)</td>
<td>11</td>
</tr>
</tbody>
</table>

Figure 9: The Performance of CPN Tools

The five single negotiation outcomes are illustrated in Figure 10 which shows utilities of five negotiations $A_1, A_2, A_3, A_4$ and $A_5$. The utilities of each negotiation are calculated by Equation (7). In Figure 10, the horizontal axis shows the negotiation time, and the vertical axis indicates the negotiation utilities. The dark blue line is the utility of negotiation
A1, which starts with utility of 0.002 ($A_1^1$) and finishes with utility of 0.406 ($A_1^2$). The red line shows the utility of negotiation $A_2$, which starts with utility of 0.135 and ends with the utility of 0.390. The purple one is negotiation $A_3$, whose utility goes up fast and the light blue one shows the negotiation $A_4$, whose utility ends up at 0.317. Negotiations $A_1$, $A_2$, $A_3$ and $A_4$ all achieved success in the end while negotiation $A_5$ failed whose utility shown in light green line is stable at the most of time. When negotiation $A_3$ failed in the end, negotiation $A_1$, $A_2$ and $A_4$ already finished. Although negotiation $A_5$ was still on-going, there was no dependence between negotiation $A_3$ and $A_5$. Therefore, the failure of negotiation $A_3$ has no impact on other negotiations.

Figure 11: Performance of five negotiations

The utilities of the MNRs $R_1$, $R_2$, $R_3$, $R_4$ and the CMN $N$ are calculated by utilizing Equation (5) and (6) in the previous Subsection 3.3. Figure 11 shows the utilities of the CMN $N$ and MNRs $R_1$, $R_2$, $R_3$ and $R_4$ where the horizontal axis shows time and vertical axis indicates utilities. The dark blue line is the utility of MNR $R_1$, which starts with utility of 0.083 ($R_1^1$) and finishes with utility of 0.264 ($R_1^2$). The red line shows the utility of MNR $R_2$, which starts with the utility of -0.052 and ends with the utility of 0.196. The light green line shows the utility of MNR $R_3$, whose utility goes up fast and eases up in the most of following time. The purple one is MNR $R_4$, whose utility goes up all the time and the light blue line shows CMN $N$ whose utility climbs first and raises to 0.226 after declining. Both MNRs $R_2$ and $R_4$ achieved success in the end while MNR $R_1$ and $R_3$ failed. Because negotiation $A_5$ failed and $R_1 = \{A_1, A_2, A_3\}$, $R_2 = \{A_2, A_4, A_5\}$, so agent $A$ terminated both $R_1$ and $R_3$ in advance in order to improve efficiency. Therefore, it can be seen that MNRs $R_1$ and $R_3$ halt at the same time in Figure 11. Likewise, the CMN $N = \{A_1, A_2, A_3, A_4, A_5\}$ achieved partial success. The agent $A$ set the MNG $\Omega_k = \frac{1}{2}$, and there are two successful MNRs $R_2$ and $R_4$ at the end of the experiment. So, agent $A$ achieved his/her goal.

The experimental results show that the proposed graph-based model and the CPN-based protocol perform CMN smoothly and effectively. Moreover, during the running of the CPN-based protocol for CMN, if some negotiation fails, agent will terminate the negotiation and the MNRs which contain the failed negotiation. This can help prediction and tradeoff to improve efficiency of CMN, especially when the CMN is seriously complicated in the real world.

6. CONCLUSION AND FUTURE WORK

In this paper, we solved some concurrent multiple negotiation problems by presenting related definitions on concurrent multiple negotiation and using a graph-based method to model concurrent multiple negotiation. Moreover, we proposed a CPN-based protocol based on the CPN to handle concurrency in concurrent multiple negotiation. The motivation of this research is to settle such a problem and handle concurrency in concurrent multiple negotiation. The experimental results showed the good performance of the proposed protocol in terms of improving efficiency of CMN.

Currently, we only conducted a case study in the experiment. In the future, we will conduct a large number of experiments to verify our proposed approach for handling the CMN.

7. REFERENCES