A Cooperative-Competitive Negotiation Model

Xuehong Tao
School of Communication and Information
Nanyang Technological University
Nanyang Ave, Singapore 639798
exhtao@ntu.edu.sg

Zhiqi Shen
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Ave, Singapore 639798
zqshen@ntu.edu.sg

Chunyan Miao
School of Computer Engineering
Nanyang Technological University
Nanyang Ave, Singapore 639798
ascymiao@ntu.edu.sg

Yin-Leng Theng
School of Communication and Information
Nanyang Technological University
Nanyang Ave, Singapore 639798
tyltheng@ntu.edu.sg

Yuan Miao
School of Engineering and Science
Victoria University
Ballarat Road, Footscray
Melbourne, VIC 8001 Australia
Yuan.miao@vu.edu.au

Yu Han
School of Computer Engineering
Nanyang Technological University
Nanyang Ave, Singapore 639798
yuhan@pmail.ntu.edu.sg

ABSTRACT

Service negotiation is a complex activity in e-business. Negotiation automation is able to free people from tedious interactions including both trivial actions, such as selection of a brand of wines for purchase, and complex tasks, such as conference organizations. Most of the existing negotiation automations are "price" bargaining type of position based negotiations, or simple alternative solution seeking type of interest based negotiations. In an e-business environment, it would be more powerful if new services could be built based on multiple parties' existing services to have a cooperative solution. This paper proposes a negotiation model to enable negotiation parties to exchange preferences and knowledge, develop optimal cooperative solutions for mutual benefits. It is a cooperative-competitive win-win strategy.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence]:** Distributed Artificial Intelligence - *Intelligent agents.*

General Terms

Algorithms, Management, Theory.

Keywords

Negotiation automation, service negotiation, interest-based negotiation, cooperative-competitive negotiation

1. INTRODUCTION

Negotiation is a key activity in e-business. E-business provides businesses with efficiency, cost saving and productivity. In the e-business environment, service consumers interact with service providers to receive services. However, in some cases, the service requested by the consumer can not be fulfilled by the producer. Hence the consumers and the producers need to negotiate their service requirements and offers.

Automated negotiation as a key type of interaction in e-business has become an increasingly popular research topic. Recently, agent technologies have been applied to automated negotiation. Negotiation automation can significantly reduce negotiation time (making large volumes of transactions possible in small amounts of time) and can also remove some of the reticence of humans to engage in negotiation (e.g., because of embarrassment or personality) [1], hence the formalization of negotiation has received a great deal of attention from the agent Community [2].

People use negotiation as a means of compromise in order to reach mutual agreement. In general, negotiation is defined as an interactive process which aims to achieve an agreement for business parties. Self interested agents work for their own goals and are competitive among each other by nature. In an e-business environment, it is also desirable for negotiation agents to have an incentive to cooperative in order to achieve efficient mutually beneficial win-win solutions. That is to say, cooperation is regarded as having the same level of importance as competition. Hence the new term *coopertition* is created to describe the cooperation-competition characteristics of business activities.

Most of the existing negotiation automations are "price" bargaining type of negotiation that focus on fixed bargaining positions, or simple interest based negotiation that focuses on seeking alternative solutions for individual agents to avoid conflicts. They are not focused on finding mutual gain solutions which will give negotiation parties an opportunity to plan on the whole (even if self interested) and make full use of all parties capabilities and maximize the overall benefit.

This paper proposes a knowledge based model for negotiation automation, and it tries to find optimal mutually beneficial solutions for the negotiation parties using shared knowledge of all parties. The rest of the paper is organized as follows. Section 2 introduces the negotiation strategies and related works. Section 3 proposes our computational model for negotiation agents. Section 4 provides the algorithms to automate the key negotiation processes, and illustrates the method with an example. Section 5 concludes the paper.

2. NEGOTIATION STRATEGIES AND RELATED WORKS

Negotiation strategies: The traditional negotiation focuses on bargaining positions, such as price, delivery time and quantity etc. It is termed *Position Based Negotiation* (PBN). If no agreement on the positions can be reached, the negotiation fails. An example for setting up an educational game environment is illustrated below and no agreement was made in this case.

A: Could you help me develop an educational game for Primary One math class?
B: Sorry, we don't have software development services.
A: That is OK. Bye.

Interest Based Negotiation (IBN) [3] focuses on satisfying the underlying reasons rather than to meet the stated demands. By discussing the reasons behind the positions and thinking of alternatives, mutually acceptable agreement is more likely to be reached. In the above scenario, the goal of A is to set up an educational game environment, so B proposes an alternative solution: buy the game system instead of developing the game system. There will be an agreement if it is acceptable to A, see example below.

A: Could you help me develop an educational game for Primary One math class?
B: Sorry, we don't have software development services.
A: I want to set up an educational game environment.
B: Do you want Math Discovery Educational Game System which is an integration of hardware, communication software and Math Discovery game software? We have it in stock.
A: That is perfect.

Cooperative-competitive Negotiation (coopertitive negotiation) is a new model of negotiation we propose in this paper that the negotiation parties can cooperatively use their knowledge to jointly create a solution acceptable to both parties. They can share information to have a more globalized view, they can exchange goals to pursue mutual benefits and share capabilities to develop cooperative solutions. Meanwhile, self interested agents work on their own benefits. They are competitive among each other. This model enables negotiators to find optimal solutions among competitive options. Hence, this is a new model of negotiation and it advanced interest based negotiation by introducing the cooperative-competitive characteristics. For the same educational game set up scenario, better solutions could be developed if it is based on multiple parties' knowledge and capabilities. As illustrated in the following example, a solution could be using the existing hardware and buying software from B with a total cost $5000, or a more cost effective solution to buy software from C and B respectively with a total cost $4500.

A: Could you help me develop an educational game for Primary One math class?
B: Sorry, we don't have software development services.
A: I want to set up an educational game environment.
B: Do you want Math Discovery Educational Game System for $8000 which is an integration of hardware, communication software and Math Discovery game software? We have it in stock.
A: I already have our hardware system.
B: You can use your hardware system and buy communication software ($2000) and Math Discovery game software ($3000) from here.
A: Ok, the total cost is $5000 and that is good.

Or another party involve in the negotiation:

A: I already have our hardware system.
B: You can use your hardware system and buy communication software ($2000) and Math Discovery game software ($3000) from here.
C: I sell communication software for $1500. You can use your hardware, buy communication software from me, and buy Math Discovery game software from B.
A: Ok, the total cost is $4500 and that is excellent.

This example demonstrated that a good negotiation strategy should exhibit the following capabilities:

- Finding alternative solutions when no agreement on stated positions.
- Exchanging information to form a globalized view.
- Choosing the optimal among competitive solutions.
- Seeking cooperative solutions that aggregate individual's capabilities.
- Pursuing mutual benefits which form the foundation of long term cooperation.

We are going to propose automated negotiation agents that are able to flexibly change negotiation positions, exchange information and preferences, hence work towards an optimal mutually beneficial cooperative solution.

Related work in agent community: Intelligent agent, as a new type of autonomous components for constructing open, complex and dynamic systems, is one of the most suitable software entities to carry out negotiation automation. Agent community also takes negotiation as a core part of agent interactions. Jennings et al. [2] defined negotiation as the process by which a group of agents try to come to a mutually acceptable agreement on some matter.

The research of negotiation automation in software agent community can be categorized into three main approaches [2]: game theoretic approach [4], heuristic approach [5] and argumentation-based approach [6][7]. The game theoretic approach applies game theory techniques to find dominant strategies for each participant. The heuristic-based approach applies heuristic decision making during the course of the negotiation. Negotiators are not allowed to exchange additional information other than the proposal in both approaches. They are mainly used for position based negotiations.

The Argumentation-Based approach allows agents to exchange additional information. It enables agents to gain a wider understanding of their counterparts, thereby make it easier to resolve certain conflicts especially for conflicts due to incomplete knowledge. Argumentation based negotiation is a broad term, it refers to all the negotiations that exchange additional meta-level information (arguments) during the negotiation process [2]. This approach provides support for interest based negotiation strategy, as negotiators can exchange their pursuing interest/goals through argumentation.

There are some recent studies using argumentation based agent approach to realize interest based negotiation strategy. To list a few, Rahwan et al [8] proposed a framework for intelligent agents to conduct interest based negotiation. They studied the relationships between agent's goals and the types of arguments that may influence other agents' decisions, as well as defined a set of locutions that can be used in the negotiation procedure. Pasquier [9] gave a fully computational specification of negotiation agents using the 3APL agent language, where the agents are able to propose alternative plan(s) for the underlying goals. Tao et al designed a computational model and algorithms to fully automate the key components of interest based negotiation [10]. Based on suitable knowledge models, automated interest based negotiation is also applied in educational contexts for curriculum negotiation [11][12]. Pasquier et al [13] conducted empirical study on interest based bargaining and reframing agents, where the agents can exchange information about their underlying interests and alternatives to achieve the interests. The simulation demonstrated the advantages of interest based negotiation.

In this paper, we propose a cooperative-competitive negotiation model. Unlike the existing interest based negotiation models, the proposed model not only uses the argumentation based approach to exchange goals or preferences, provides alternative solutions to avoid conflicts but also introduces cooperative and competitive characteristics.

More specifically, our model distinguishes itself from the existing interest based negotiation in the following aspects. Firstly, most of the existing interest based negotiation models focus on individual alternative solution seeking so as to avoid conflicts. Our model focuses on multiple party joint solution construction to resolve the conflicts. It is a cooperative solution. Secondly, the methods in existing interest based negotiations are to find a solution without conflict. Our model is able to find the optimal solution during the process of searching for non-conflicting solutions. It is a competitive solution. Thirdly, some existing methods have restrictions that higher level goals (from the same agent or different agents) cannot share sub goals or resources, so as to remove the potential conflict. They are more suitable to model agents that work separately and in separate domains. Our model also allows agents to share sub-goals and resources, and enables agents (even if self interested) to build solutions that satisfy the combined goals from multiple parties. Overall, our model advances the existing interest based negotiation methods by introducing the cooperative competitive characteristics.

3. COOPETITIVE NEGOTIATION AGENT

3.1 Overview

Agents are autonomous entities that make decision independently and work towards their goals. Complex goals can be considered as a composition of sub goals. Sub goals may be further decomposed to next level sub goals. The goals and the sub goals form a hierarchical structure. The goals and their relationships are the knowledge of agents to interact with the environment and evolve. The knowledge is maintained in the knowledge base of agent.

The Coopertitive Negotiation Agent proposed in this paper is a generic model representing the core parts of cooperative-competitive negotiation. The main components are a knowledge base and a negotiation engine.

The knowledge base stores the knowledge about goals. The negotiation engine manages the negotiation process and generates negotiation solutions automatically. It has the following main functionalities:

- Generate a Proposal: In the context of e-Business context, for service provider, a proposal is an offer to consumers for certain services. For service consumer, a proposal is a request for certain services.
- Accept/Reject a Proposal: Whether to accept or to reject a proposal depends on many factors, including whether a consumer needs the offer, whether the provider is able to offer the service and whether the price, time, quality or other criteria are satisfied.
- Exchange Information: An agent normally has incomplete knowledge. So the decision is made based on limited local information. If agents exchange information during the negotiation, it is possible to find more options for solving a problem. Hence there are more chances to achieve an agreement.
- Develop a Mutual Beneficial Solution: Agents have the ability to make use of information shared from other agents, find a solution to meet goals of all agents.
- Alter Negotiation Positions: If no agreed deal is reached, an agent may consider changing to other sub goal(s) while still supporting the same super goal.

3.2 Knowledge Model for Coopertitive Negotiation Agent

In e-business environment, a negotiation agent should have knowledge about its goals and how complex goals can be composed from elementary goals where the elementary goals can be achieved by primary services. The knowledge base of a negotiation agent is a collection of goals and relationships among goals. It is defined as a 3-tuple $KB = \langle G, R, C \rangle$, where

$$G = \{ g_i \mid i = 1, 2, \dots n. \}$$
$$R = \{ r_i: g_{i0} \rightarrow g_{i1}, g_{i2}, \dots g_{ik} \mid g_{i0}, g_{i1}, \dots g_{ik} \in G, i = 1, 2, \dots m\}$$
$$C = \{c(g) \mid g \in G\}$$

G is a goal set, R is a relationship set where each relationship r_i describes how a super goal is decomposed to sub goals. g_{i0} is termed as the head of a relationship, $g_{i1}, g_{i2}, \dots g_{ik}$ are termed as the tail of a relationship.

C is a criteria set which will be discussed later. $c(g)$ is the criteria values relevant to g, such as price, delivery time, quality of service, payment methods and etc.

According to the super-sub goal relationship, goals of an agent form a goal hierarchy, which is a network and it is not necessary a tree.

- **Atom Goal**

A goal g is called an atom goal if there is no decomposition relationship such that it has g as the head and other goals as the tail. Atom goals are goals that can not be decomposed to other sub goals. They are corresponding to the primary services in an agent's belief.

An atom goal of one agent maybe a composite goal of another agent, because agents have different belief about the basic services they can operate. For example, for a real estate agent, obtaining a house is an atom goal. However it is a composite goal for a builder agent which may contains a sub goal of buying a block of land and a sub goal of building a house.

- **Decomposition**

Following some relationship in R, a goal g can be decomposed into sub goals (not necessarily atom goals). The set of the sub goals are called a decomposition of g. A goal may have different decompositions.

A goal is achievable if it can be decomposed to a set of atom goals, and the services corresponding to the atom goals are all available.

For example, in a holiday booking scenario,

G ={ g_1 = "have holiday booking",
g_2 = "have transport booking",
g_3 = "have accommodation booking",
g_4 = "obtain air ticket from X Airline",
g_5 = "obtain booking of A Hotel",
g_6 = "obtain train ticket from Y railway services" }

R= {r_1: $g_1 \rightarrow g_2, g_3$, r_2: $g_2 \rightarrow g_4$, r_3: $g_3 \rightarrow g_5$, r_4: $g_2 \rightarrow g_6$ }

Here, {g_2, g_3}, {g_4, g_3}, {g_4, g_5} and {g_6, g_5} are all decompositions of g_1. Goal g_1 can be achieved by { g_4 , g_5} or { g_6 , g_5} , i.e. for a holiday booking, one solution is to take flight of airline X and live in Hotel A. Another solution is to go by train from Y Railway services and live in Hotel A.

- **Criteria of Goals**

There are some criteria to describe a goal (service), such as price, delivery time, quality of service, payment methods and etc. We define the criteria of a goal g as a vector (v_1, v_2, … , v_n) from a domain vector (D_1, D_2, …, D_n).

$$c(g) = (v_1, v_2, \dots , v_n) \in (D_1, D_2, \dots, D_n), D_i \text{ is the domain of } v_i.$$

For example, if a goal g is "Buying a Lenovo Notebook model S10". $c(g)$=($900, 2, {cash, credit card}) from domain (R^+, I^+, {cash, credit card, bank transfer}). This may mean, the price is $900 from a positive real number domain, the delivery time is two days from a positive integer domain, and the payment method is either by cash or by credit card from a set domain contains all possible payment methods.

For a certain service, the values in the criteria allow negotiators to make comparison between competitive solutions and to request an optimal one. Suppose agents are able to compare the preference among multi-criteria [14]. For example a simple way could be by using weight to combine all dimensions in the criteria to a single value then compare this single value.

In the rest of this paper, we consider criteria as a single value and suppose the smaller value is the better without loss of generality. For composite goals, they have different decompositions each having different criteria values. c(g) is the smallest among them or a lower bound of them. The estimated criteria of composite goals can be used as a heuristic in search algorithms. Choosing a small estimated value can make sure the goal has more opportunity to be considered. For atom goals, if it corresponds to an available service, c(g) is the actual service criteria value. If it is corresponding to an unavailable service according to the agent's knowledge, c(g)= +∞.

- **AND/OR Graph Representation of Knowledge Base**

For easy presentation of our algorithms, we also define the graph representation of a knowledge base. An AND/OR Graph [15] is a hyper graph. Instead of arcs connecting pairs of nodes in the graph, there are hyper arcs connecting a parent node with a set of successor nodes. These hyper arcs are called connectors. Suppose KB=<G_{KB}, R, C> and its AND/OR Graph representation is Q=(G_Q, E, C), where

$G_Q = G_{KB}$, i.e. nodes in Q are the goals in KB,

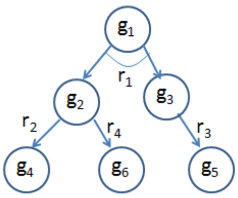
E={ (g_{i0} , { g_{i1}, g_{i2}, … g_{ik} }) | $g_{i0} \rightarrow g_{i1}, g_{i2}, \dots g_{ik} \in R$}, i.e. connectors in Q are decomposition rules in KB.

Leaf nodes in Q are atom goals in KB.

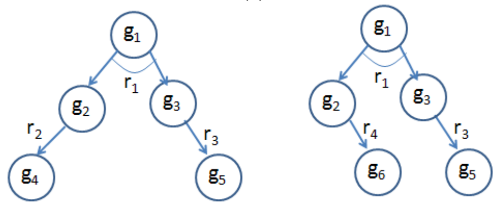
- **Solution Graph and Partial Solution Graph**

In an AND/OR graph Q, a node g can be expanded to its successors by following exactly one connector. Each successor node can be expanded further in the same way and a graph rooted on g will be generated. The graph is called a Partial Solution Graph of g. If all the leaves of the partial solution graph are the leaves of Q, the partial solution graph is a solution graph. Partial solution graph and solution graph are graph representations of goal decompositions.

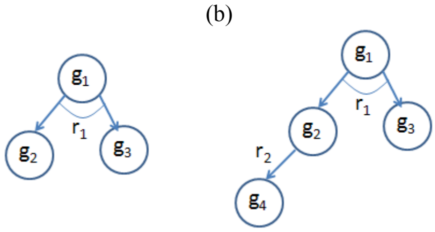
In the above holiday booking example, the AND/OR Graph representation of the knowledge base is shown in Figure 1(a). Two possible solution graphs are shown in Figure 1(b) and two partial solution graphs are shown in Figure 1(c).



(a)



(b)



(c)

Figure 1. Graph representation of the holiday booking KB
(a) Graph representation of the knowledge base (b) Possible solution graphs (c) Partial solution graphs

Suppose the knowledge base of an agent is maintained periodically so that it has no loop decomposition and the decompositions are all minimal. The requirement of non loop decomposition means a goal's decomposition can not include the goal itself. Formally, there is no decomposition *Z* of a goal *g* such that $g \in Z$. Minimal decomposition means there is no decompositions Z_1 and Z_2 of a goal *g* such that $Z_1 \subset Z_2$. i.e. the rules will not produce unnecessary sub goals. For example, if $\{g_1, g_2\}$ and $\{g_1, g_2, g_3\}$ are two of the decompositions of a goal, then it does not meet the minimal decomposition requirement because g_3 is unnecessary.

Knowledge Base Revision [16] can provide the system with learning capabilities by adding in new knowledge and removing/revising existing knowledge during the negotiation process. The details of knowledge base revision will be omitted here.

4. NEGOTIAITION AUTOMATION

- **Goal Decomposition Algorithm**

Firstly, we will provide a method to decompose a goal, named *g*, to atom goals (which correspond to primary services) using a heuristic search strategy.

Suppose we have a knowledge base KB which contains relationships about goal decompositions. For an atom goal, if it corresponds to an available service, c(g) is the actual service criteria value. If it is not available, c(g)= $+\infty$. Suppose the agent is able to perform multiple criteria preference analysis [14] and find the solution with the optimal criteria. For simplicity, we consider the smaller criteria solution as the better one.

Algorithm Decompose listed below will decompose *g* to atom goals based on Nilsson's AO* algorithm [15]. During the process of creating a search graph and marking a partial solution graph, the algorithm is gradually approaching to the optimal solution by using the criteria of each goal as heuristics. The algorithm starts from *g*, selects and marks the connector with the smallest criteria as the temporary best solution for g. Then continues to decompose the sub-goals of g. Whenever new information that makes changes to the criteria of a goal is encountered, the algorithm will propagate the newly discovered information up the goal hierarchy, re-calculate the criteria and make a new selection among connectors.

Algorithm. Decompose (*g*)

1. Create a search graph Q, Q ={ *g* }
   If *g* is an atom goal, label *g* as *Solved*. cost (*g*) = c (*g*)
2. Until *g* is labeled *Solved* , or cost (g) = $+\infty$ do
   a. // Select node to expand
      Compute a partial solution graph H in Q by tracing down marked connectors in Q from *g* (marks will be discussed later in this algorithm)
      Select any non terminal leaf node *n* of H
   b. // Expand node *n* by generating its successors
      - If $n \rightarrow n_1, n_2 \ldots n_k \in R$, Add all sub goals of *n* to Q
      - For successors n_j not occurring in Q, cost(n_j)= c(n_j)
      - If n_j is leaf, label *Solved*.
   c. // Propagate the newly discovered information
      // up the graph
      S={*n*} // *S* is a set of nodes that have been labeled
      // solved or whose cost have been changed
      Until *S* is empty do
      - Remove a node *m* (*m* has no descendants in S) from S
      - //Computer the cost of each *m*'s decomposition
        // cost (*m*) is the minimum cost among all connectors
        For each connector $m \rightarrow m_{i1}, m_{i2}, \ldots, m_{ik}$
        $Cost_i(m) = cost(m_{i1})+cost(m_{i2})+\ldots+cost(m_{ik})$
        $Cost(m)=min_i (cost_i (m))$
        Mark the best path out of *m* by marking the connector with minimum cost
      - If all nodes connected to *m* through this new marked connector has been labeled *solved*, label *m solved*
        If *m* solved or cost of *m* just changed, add all of the ancestors of *m* to *S*
3. If *g* is labeled *Solved*, return *True*, else return *False*

End of Decompose.

- **Proposal Generation**

An agent selects its high level goal, named *g* based on certain reasoning mechanism. If algorithm Decompose (*g*) returns *True*, the partial solution graph H is the current pursuing solution graph for goal *g*. Based on H, if a goal can not be realized by the agent itself, it will be proposed to other agents. A proposal could be an offer proposal from the provider agent to advise its services, or a request proposal from the consumer agent to ask for services.

Hence a proposal is a goal $g \in H$. It can be an atom goal for a single service, or a composite goal for a complex service.

- **Cooperative-Competitive Solution Construction**

When an agent receives a proposal *g*, it will evaluate it and then decide whether to accept or deny it. If no agreement can be reached, the participating agents may consider exchanging negotiation related information, including information from KB and pursuing goals.

Upon receiving new knowledge from other agent(s), the agent will carry out a temporary knowledge base revision by adding the new knowledge to its existing knowledge base. Whether to incorporate the new knowledge permanently in the knowledge base will be decided by the agent through other mechanism. The temporary knowledge base revision can be implemented by algorithm KBRevision listed below.

Suppose the knowledge base of the agent is KB=<G, *R, C*>, and the agent will revise the KB to incorporate new knowledge noted as KB'=< G', *R', C'* >.

Algorithm. KBRevision ( )

For each new goal in G', add into G

For each new relationship in R', add into R if it doesn't cause loop decomposition

For each new criteria $c_{new}(n)$
   If there is no criteria of n exists in KB, add $c_{new}(n)$ into C
   If there is criteria $c_{old}(n)$ exists and $c_{old}(n) \neq c_{new}(n)$,
   a. $c(n) = min(c_{old}(n), c_{new}(n))$, which makes sure the low criteria solution has the opportunity to be selected.
   b. propagate the new criteria to upper lever goals (details will be omitted here as it is similar as what have been done in algorithm Decompose, step 2.c.)
   c. If n is an atom in KB'
      Add $n \rightarrow n'$ in KB, $c(n')=c_{new}(n)$
      If *n* is an atom in KB

Add n→n" in KB, c(n")=c_{old} (n)

End of KBRevision.

Based on the newly build temporary knowledge base,

If Decompose (*g*) =*True*

Partial solution graph H is the solution to *g*

This solution is a cooperative solution because it is constructed on both parties' available options. It is also a competitive solution because it selected the best cost solution.

- **Mutual Beneficial Solution Construction**

If party A has goals g_A^1, g_A^2, … g_A^s and party B has goals g_B^1, g_B^2, … g_B^t, they want to seek opportunity to achieve their mutual goals. We can add decomposition knowledge $g_{Mutual} \rightarrow g_A^1, g_A^2, \ldots g_A^s, g_B^1, g_B^2, \ldots g_B^t$, into the knowledge base. If Decompose (g_{Mutual}) is *True,* the partial solution graph H is the solution to g_{Mutual}.

- **Negotiation Position Alternation**

If there is no solution for the current proposal *g*, the participating agents may also consider other alternative goals that support the same super goal as that *g* does. This can be achieved by

f = father of *g* in the current pursuing solution graph G

make *f* the new proposal

By doing so, the agent changes the negotiation position from *g* to *f,* and work on other possibilities to achieve *f.*

- **Correctness and Advantages of the Method**

If no solution for *g*, i.e. all decompositions of *g* contain unavailable services, according to the algorithm cost(*g*) will reach +∞, so the algorithm returns false.

If there is a solution from *g* to a set of atom goals, and if for all goal decomposition relationship $n \rightarrow n_1, n_2 \ldots n_k$, $c(n) \le c(n_1) + c(n_2) \ldots + c(n_k)$, the algorithm will terminate and return True. By tracing the marks, graph H is the optimal solution. cost(g) is the cost of the solution.

Hence, with the restriction that for all composite goal g, the estimated criteria c(g) is always smaller than the sum of its sub goals, i.e. the estimated criteria is always smaller than the real criteria, the algorithm can find the optimal solution.

By limiting the estimated criteria of a goal *g* to be not bigger than the actual criteria, the actual low criteria solution of *g* will have the opportunity to be explored. However, if the estimated criteria are much lower than the actual criteria, this will direct the algorithm to spend time to explore this seemingly optimal but actually not optimal branch. Hence a good estimation will reduce the unnecessary search and find the optimal solution.

The proposed method is flexible in handling negotiation conflicts and has the following advantages:

- Find alternative solutions or alter pursuing goals when there is no agreement on initial negotiation positions.
- Find cooperative solutions based on the knowledge of multiple parties.
- Find optimal solutions among competitive options.
- Find mutual beneficial solutions by using a joint goal.

- **Example**

We are going to use a simple example to illustrate the proposed cooperative-competitive negotiation strategy.

Suppose AB University (ABU) wants to organize a conference. The agent A_1 of ABU negotiates with the agent A_2 of Event Management Company (EOC) for relevant services. For simplicity of presentation, we define some symbols to represent the goals. Suppose

g_1 : Organize conference
g_2 : Self-organize the conference
g_3: Arrange meeting room
g_4 : Print meeting materials
g_5 : Arrange museum visit
g_6 : Rent room from CD Hotel
g_7 : Use AB University meeting room
g_8 : Operate business
g_9: Out source conference management (ABU) / Provide conference management for others (EOC)
g_{10} : Manage celebration activity
g_{11} : Arrange city tour
g_{12} : Rent meeting room from EF Centre

Suppose the knowledge base of A_1 is KB_1 and the knowledge base of A_2 is KB_2. For simplicity, we put the (estimated) price with the goal together.

$KB_1=(G_1, R_1, C_1)$ where

$G_1(C_1)=$ {g_1(\$8000), g_2(\$8000), g_3(\$0), g_4(\$3000), g_5(\$5000), g_6(\$3000), g_7(\$0)}

$R_1=$ {$g_1 \rightarrow g_2$; $g_2 \rightarrow g_3, g_4, g_5$; $g_3 \rightarrow g_6$; $g_3 \rightarrow g_7$; }

$KB_2=(G_2, R_2, C_2)$ where

$G_2=$ {g_1(\$9000), g_3(\$3000), g_4(\$4000), g_6(\$3000), g_8(\$9000), g_9(\$10000), g_{10}(\$8000), g_{11}(\$3000), g_{12}(\$4000) }

$R_2=$ {$g_1 \rightarrow g_9$; $g_8 \rightarrow g_9$; $g_8 \rightarrow g_{10}$; $g_9 \rightarrow g_3, g_4, g_{11}$; $g_3 \rightarrow g_6$; $g_3 \rightarrow g_{12}$; }

The current goal of A_1 is to "organize conference". After calling Decompose (g_1), the solution graph is listed in Figure 2 (the price is listed beside each goal node). A_1 proposes to use its meeting room with no cost (g_7), print meeting materials by itself (g_4) and request others to arrange the museum visit (g_5). The total criteria is about \$8000.

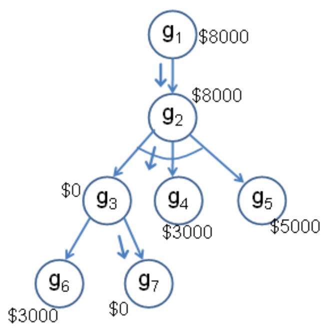


Figure 2. Solution Graph of g_1 in A_1

There is no service relevant to "arrange museum visit (g_5)", A_2 rejected the proposal. No agreement on the initial proposal, A_1 will consider altering the initial negotiation position. A_1 will share its goal "self organize conference (g_2)". A_2 still has no relevant services. A_1 will continue to share its higher level goal "organize conference (g_1)".

With the knowledge that A_1 is aiming to organize the conference, A_2 knows that "organize conference" can be done by not only "self organize the conference" ($g_1 \rightarrow g_2$) but also "out source conference management" ($g_1 \rightarrow g_9$). A_2 is able to "provide conference management for others" (g_9), so it provides an alternative solution to A_1 that A_2 will help A_1 to organize the conference and replace the "arrange museum visit (g_5)" with "arrange city tour (g_{11})". The solution graph (by tracing down the marks from g_1) is listed in Figure 3. The total cost is \$10000. Because the algorithm only expands the relevant nodes, goals such as $g_8$and g_{10} are not considered here.

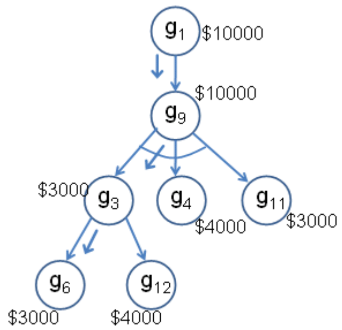


Figure 3. Solution Graph of g_1 proposed by A_2

With the relevant knowledge shared by A_2, A_1 could revise its knowledge base to incorporate the new knowledge KB'=<G', R', C'> contained in A_2's proposal, where G'={g_1, g_9, g_3, g_4, g_{11}}, R'={$g_1 \rightarrow g_9$; $g_9 \rightarrow g_3$, g_4, g_{11};}, C'={c(g_4)=\$4000, c(g_{11})=\$3000, c(g_9)=\$10000}. After using algorithm KBRevision to incorporate the new information, the temporary knowledge base of A_1 is as shown in Figure 4.

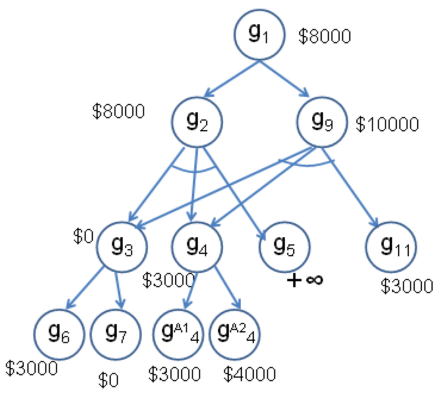


Figure 4. The temporary knowledge base of A_1

From the temporary knowledge base, A_1 could use Decompose (g_1) to build a solution graph as shown in Figure 5. The total cost is \$6000.

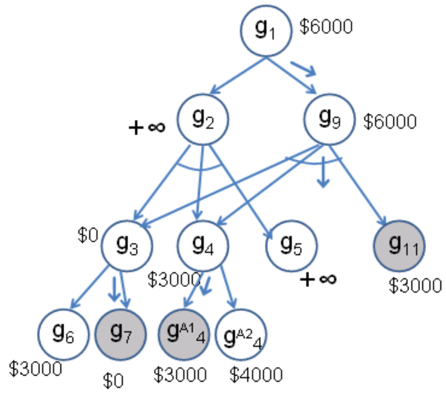


Figure 5. Solution Graph of g_1 based on shared knowledge

If there is a Tourism Company (TC), whose agent A_3 shares knowledge about its service "arrange city tour (g_{11})" with the cost of \$2000, a more cost effective solution could be built as shown in Figure 6. The total cost is \$5000. The final solution constructed is a cooperative solution from three parties and with the best cost among the competitive options.

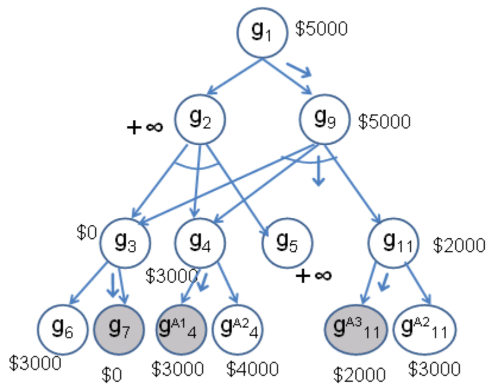


Figure 6. Three Parties Cooperative Solution Graph of g_1

As it shows, most of the current interest based negotiations focus on individual alternative solution seeking, whereas our model is able to build alternative multi-party joint solutions and choose the most effective one.

5. CONCLUSION AND FUTURE WORK

This paper proposed a new computational model for negotiation automation: cooperative-competitive negotiation. As cooperative-competitive negotiation allows involved parties to dig into the higher level goals behind their positions, use mutual knowledge to construct new solutions. The solutions are planned based on knowledge and preference from all parties, which is a cooperated mutually beneficial decision. The cooperative-competitive negotiation is more powerful and constructive than position based negotiations or simple alternative solution seeking kinds of interest based negotiations.

In our subsequent research, we will focus on the design of knowledge models (such as using Fuzzy Cognitive Map [17], Dynamic Cognitive Networks [18]) that better represent human negotiation processes.

6. REFERENCES

[1] Lomuscio A.R., Wooldridge M., Jennings N.R. 2003. A Classification Scheme for Negotiation in Electronic Commerce. Int. Journal of Group Decision and Negotiation, Vol 12, No 1, January 2003 , 31-56

[2] Jennings, N.R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., Wooldridge, M. 2001. Automated negotiation: prospects, methods and challenges. Int. Journal of Group Decision and Negotiation, Vol. 10, No. 2 , 199–215

[3] Fisher, R., Ury, W. 1983. Getting to Yes: Negotiating Agreement without giving in. Penguin boks, new York

[4] Steven J. Brams. 2003. Negotiation Games: Applying Game Theory to Bargaining and Arbitration. Published by Routledge

[5] Faratin, P., Sierra, C., Jennings, N.R. 2002. Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. Journal of Artificial Intelligence, Vol. 142, No. 2, 205-237

[6] Karunatillake, N.C., Jennings, N.R., Rahwan, I., Norman, T.J. 2005. Argument-based negotiation in a social context. Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 1331-1332

[7] Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L. 2004. Argumentation-based negotiation. The Knowledge Engineering Review, Vol.18, No.4, 343-375

[8] Rahwan, I., Sonenberg, L., Dignum, F. 2004. On Interest based Negotiation. Lecture Notes in Computer Science, Springer-Verlag, Vol. 2922, 383-401

[9] Philippe Pasquier, Frank Dignum, Iyad Rahwan and Liz Sonenberg. 2006. Interest-Based Negotiation as an Extension of Monotonic Bargaining in 3APL. Lecture Notes in Computer Science, Volume 4088, 327-338

[10] Xuehong Tao, et al. 2006. Interest Based Negotiation Automation. Proceedings of International Conference on Intelligent Computing, Kunming, China, August, 2006. Lecture Notes in Computer Science (Computational Intelligence and Bioinformatics), Volume 4115, 211-222

[11] Xuehong Tao, Nicola Yelland, Yuan Miao. 2008. Adaptive Learning through Interest Based Negotiation. the 16th International Conference on Computers in Education, ICCE 2008, Taiwan, October

[12] Xuehong Tao, Yuan Miao. 2008. Interest Based Learning Activity Negotiation. Proceedings of Cyberworlds 2008, HangZhou, China, September

[13] Philippe Pasquier, Ramon Hollands, Frank Dignum, Iyad Rahwan, Liz Sonenberg. 2007. An empirical study of interest-based negotiation. Proceedings of the ninth international conference on Electronic commerce, 339 - 348

[14] Hoh Peter In, David Olson. 2004. Requirements Negotiation Using Multi-Criteria Preference Analysis. Journal of Universal Computer Science, Vol. 10, No. 4, 306-325

[15] Nilsson, Nils J. 1982. Principles of Artificial Intelligence. Published by Birkhäuser.

[16] Eiter, T., Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence, Vol. 57, No. 2-3, 227-270

[17] C. Y. Miao, Z.Q. Liu, Y. Miao, and A. Goh. 2003. DCM: Dynamical Cognitive Multi-Agent Infrastructure for Large Decision Support System, International Journal of Fuzzy Systems, Vol. 5, No3, 184-193

[18] Y. Miao, Z.Q. Liu, C. K. Siew, C.Y. Miao. 2001. Dynamic Cognitive Net-an Extension of Fuzzy Cognitive Map, IEEE Transaction on Fuzzy Systems, Vo19, No5, 760-770