

Association for Information Systems

**AIS Electronic Library (AISeL)**

---

ICEB 2002 Proceedings

International Conference on Electronic Business  
(ICEB)

---

Winter 12-10-2002

## Overcoming the Fixed-Pie Bias in Multi-Issue Negotiation

Raymund Lin

Seng-Cho T. Chou

Follow this and additional works at: <https://aisel.aisnet.org/iceb2002>

---

This material is brought to you by the International Conference on Electronic Business (ICEB) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICEB 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Overcoming the Fixed-Pie Bias in Multi-Issue Negotiation

Raymund Lin

Institute of Information Management  
National Taiwan University  
Taipei, Taiwan  
d85725004@ntu.edu.tw

Seng-Cho T. Chou

Institute of Information Management  
National Taiwan University  
Taipei, Taiwan  
chou@im.ntu.edu.tw

## Abstract

Multi-issue negotiation may produce mutual beneficial results to both negotiators while single-issue negotiation can not. However, there are difficulties in automating a multi-issue negotiation, since the search space grows dramatically as the number of issues increases. Although many concession strategy learning mechanisms have been proposed to deal with the problem, recent research uncovered that the fixed strategy of concession and the fixed-pie bias are the two major interferences in the automation of multi-issue negotiation. It is suggested that the lack of communication between agents may have impeded information sharing and joint-problem solving possibilities.

In this paper, we show that the fixed-pie bias can interfere with the negotiation outcome if there are non-conflicting issues. We propose a new negotiation model and an innovative algorithm that not only allows information to be shared in a controlled way, but also allows the information shared to be effectively used for conducting a systematic search over the negotiation problem space. The combined mechanism is capable of using strategies learned from counter-offers and is immune to the fixed-strategy limitation and the fixed-pie bias. It contributes to the automation of multi-issue negotiation in the context of open and dynamic environments.

## 1. Introduction

Negotiation is a human behavior existed since there is trade. People exchange products in order to give out what they have for what they need. In the age of electronic commerce, negotiation has been automated by software agents or supported by negotiation support systems (NSS). Research [10] [16] [11] [5] indicated that these automation mechanisms were able to reduce significantly the negotiation time and alleviate the negative effects of human cognitive bias and limitation. The evolution went from single-issue negotiation to multi-issue negotiation. In a single-issue negotiation, the term negotiated is limited to price. Although this kind of negotiation works for some cases such as auction houses, most companies on the Internet are generally against it since it brings out price wars that not only causes chaos in markets but also ignores the importance of other issues such as warranty period and delivery time.

Multi-issue negotiation becomes an important research area in the e-commerce domain, since it is more

beneficial comparing to single-issue negotiation [14]. However, there are difficulties in automating a multi-issue negotiation. Take a bilateral multi-issue negotiation, where there is a buyer and a seller performing a one-to-one negotiation, for example. Issues can be negotiated sequentially or simultaneously. The issue-by-issue approach suffers the drawback of being unable (or costly) to go back to already negotiated issues. Hence it is inappropriate for solving problems with inter-dependent issues. In the simultaneous approach, on the other hand, negotiators get lost easily in the complex decision tree of concessions, and the search space grows dramatically as the number of issues increases. Since we assume self-interested agents, an agent will not disclose his utility function, being afraid of the fact that the opponent will take advantage of it to squeeze surplus out of him. Given the situation, the only information disclosed are the offers proposed and the information of acceptance or rejection on the proposed offers. An algorithm is required to search the negotiation problem space for mutual beneficial agreements base on the limited information. Adopting the simultaneous approach, there are three types of algorithms:

1. Brute force: suppose that a buyer is bidding for goods from a seller. He then queries the seller for acceptance of each offer in mind, from the one with highest utility to the one with lowest. The seller may accept it or reject it. This process continues until the buyer has run out of alternatives. To speed up the process, the seller can counter-offer using the same method as the buyer. Once an offer is accepted by one of two players, the negotiation ends. This approach is similar to the continuous double auction used in Kasbah [2], and may work fine in the single-issue negotiation without time constraints (i.e. deadline, bargaining cost, etc.). However, it is hardly applicable to the multi-issue negotiation since there are generally too many alternatives generated by combining options of all the issues. Simply generating and sorting them will require a lot of memory space, not to mention querying them one by one, which can be very time-consuming.

2. Use a concession strategy learned from past negotiations: this approach works great if the market is static, which means that the general utility function of buyers/sellers does not change radically over time. Although we do not know exactly the utility function of the opponent, we learn from past negotiations to get the best concession strategy that can both speed up the negotiation and get better outcomes. However, if the

market is dynamic, this fixed strategy may not achieve mutually beneficial outcomes each time, since different opponents may have quite different utility functions over the various issues. Research [13] [12] [17] belongs to this type.

3. Use a concession strategy learned from counter offers: although rarely seen in the research, it is possible to learn a concession strategy from counter offers within a negotiation session. The rationale behind it is to find out which terms are important (weighted more in the utility) to the opponent, but less important to us. Conceding on these issues increases the possibility of being accepted by the opponent, while preserving our interests. Research [4] uses similarity criteria to make issue trade-offs; it is believed that by making an offer similar to the one by the opponent can approximate the preference structure of the opponent.

In an open and dynamic environment, such as the Internet, the third type algorithm becomes important. Goh et al. [6] conducted an experimental study on computer-supported bargaining in the context of electronic commerce, and uncovered that the fixed strategy of concession and the fixed pie bias are the two major interferences in the automation of negotiation. It is suggested that the lack of communication between agents may have impeded information sharing and joint-problem solving possibilities. The problem of fixed strategy has been addressed later in [4] (though the existence of too many assumptions has caused their research to be less applicable to real cases). However, the joint-problem solving possibilities are still constrained by the fixed-pie bias and the information shared.

In this paper, we propose a new negotiation model and an innovative algorithm that not only allows information to be shared in a controlled way, but also allows the information shared to be effectively used for conducting a systematic search over the negotiation problem space. The combined mechanism is capable of using strategies learned from counter-offers and is immune to the fixed-strategy limitation and fixed-pie bias. It contributes to the automation of multi-issue negotiation in the context of open and dynamic environments.

The sections are organized as follows: section 2 firstly explains the rationale for information sharing. Section 3 describes the negotiation model used, and section 4 gives details of the searching algorithm. Section 5 contains experimental analysis and finally the conclusion and future work is presented in section 6.

## 2. The Rationale for Information Sharing

In the context of Game Theory, the two-player bargaining game is defined as a non-cooperative game where two players attempt to divide a good, say a pie, between them. However, the pie-dividing concept may introduce the fixed-pie bias, which is a tendency for negotiators to assume that their own interests directly conflict with those of the other party [1] [18]. The problem caused by this bias seldom occurs in a bargaining since

most of them are dealing with conflicting issues. Let us consider a more general two-player multi-issue negotiation problem, where the players not only negotiate for the price, but also a date for delivering the product. The utility gained from settling for a different date is illustrated in Figure 1. At the beginning, their preferences seemed to be conflicting; nevertheless, they managed to meet at day 7, which is the best solution for both of them. The  $f$  curve occurs when a player wishes to get the product immediately or on a later date if he can not get it immediately. This example illustrates that sometimes the pie-dividing concept may be misleading, causing both negotiators to neglect a mutually beneficial solution if they do not communicate. Same problems can occur when dealing with qualitative issues such as the choice of colors, since the utility gained by each color may not always be conflicting with the opponent's preference.

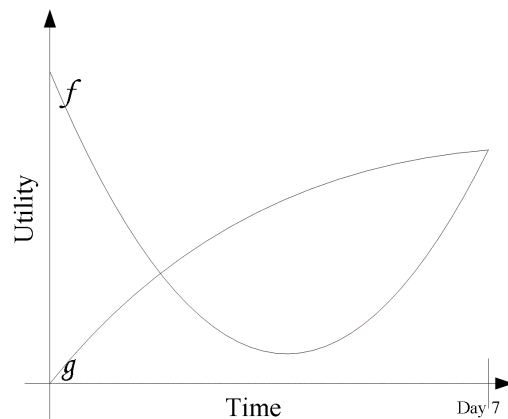


Figure 1. Non-conflicting utility case

Research [4] was based on the pie-dividing concept; however, they allow different levels of importance to be attached to various negotiation decision variables. This makes the negotiation a non-zero-sum game, where players can find mutually beneficial agreements by making trade-offs over issues instead of conceding at the utility. That is to say, by increasing some decision variables in value and decreasing some others may create an offer that will benefit one or both of the players simultaneously. However, considering the above delivery time case, we find that a mutually beneficial offer can be found by simply increasing the utility of one of the decision variables without decreasing any other (instead of conceding, one can raise the utility of the new offer). In other words, better solutions can be found if we avoid the fixed-pie bias. Nevertheless, by avoiding the fixed-pie bias, we are again lost in the enormous possible paths of solution searching. It is our belief that, by disclosing a little more information, the player with less privacy concerns can improve the negotiation outcome, and therefore benefits himself. And it is also common that, in the real life situation, a player with less privacy concerns will simply release trade-off information, such as “you can deliver the product to me immediately or a week later.”

One may argue that, in the above delivery date case, the opponent will propose the offer of day 7 eventually if you did not propose it, yet in the multi-issue case, there can be a lot of alternatives that it is too hard to enumerate them all. Therefore how the information can be shared in a systematic way such that the opponent can use it to improve the solution searching process is the major concern here. Besides, the information should be disclosed in a controlled way so that it not only matches our privacy preference but also complies with our negotiation strategy. To tackle this problem, we propose a new negotiation model and an innovative solution to the searching problem in the following sections.

### 3. Tagged Multi-offer Negotiation Model

In this section we first provide an overview of our negotiation model. Then we discuss the issues of time constraints and information states.

#### 3.1 The Negotiation Model

The settings of our negotiation model are based on a non-cooperative, multi-stage, incomplete-information, and two-player bargaining game. Since this is a non-cooperative game, each player does his best to maximize his own interest and will not share information more with the other than necessary. We assume each player knows neither the other's preferences on issues, nor the utility function; therefore it is a game with incomplete information. Given such a situation, in order to reach an agreement that is beneficial to both players, the negotiation will continue for more than one stage.

Rubinstein proposed a multi-stage bargaining game [15], in which each player of the game proposes an offer in turn and the other may accept or reject it. The game will continue until an offer is accepted. Our model differs from [15] in that: Firstly, we allow multiple offers to be proposed at one time. For example, a seller can propose two offers at one time, saying "you can buy product A at price B with delivery time C or buy product A at price D with delivery time E". The buyer may counter propose with sayings like "I can only buy product A at price F with delivery time G or buy product A at price H with delivery time K". This relaxation of multiple alternating offers enables a player to disclose possible trade-off information over which the opponent can consider. It can speed up the searching process and avoid falling into local optima in the solution landscape too early.

Besides the relaxation of multiple alternating offers, we allow players to tag information of their preferences over proposed offers. For example, a vector  $\langle\langle A, 3 \rangle, \langle B, 2 \rangle, \langle C, 1 \rangle\rangle$  indicates that offer A is the most preferred, offer B is the second most preferred, while offer C is the least preferred. We allow the multiple alternating offers and the tagging of preference information because we believe the information may be "necessary" for both players to reach an agreement that is beneficial to both of them. However, how much information will be shared is a

decision made by the players. An elegant controlling mechanism is embedded in our searching algorithm (detailed in the next section), and it can be tuned according to the privacy preference of the player.

Finally, the negotiation game will end in a known period of time, no matter whether an agreement has been reached. If an agreement has been reached before the deadline, the players will continue to improve it until the deadline. If no agreement has been reached before the deadline, the negotiation fails. If more than one agreement has been reached, the latest one is chosen. This design permits better solutions to be found after the first agreement has been reached.

We name this negotiation model a TAMON (TAGged Multi-Offer Negotiation) model.

#### 3.2 Time Constraints

Rubinstein's model has a basic assumption that 'time' is valuable during the negotiation, and the fixed bargaining cost or fixed discounting factor may affect the strategy used in the bargaining. If each player has complete information about the preferences of the other, a weaker player (i.e. the one with higher fixed bargaining cost) will always lose. However, assuming each player does not know the preference structure of the other (incomplete information), the weaker player may try to cheat the other player by making the other player believe that he is actually stronger.

To simplify the TAMON model (avoiding the consideration of time-related strategies), we assume the cost of time is negligible during a TAMON game. In other words, by choosing a reasonable period of time to play the negotiation game, it is possible for both players to agree that there can be no time constraints during the period of negotiation. By this simplification we make TAMON a micro negotiation game. It does not mean that there are no time constraints any more, but that the time constraints are omitted during the TAMON game. The time constraints still exist between the current TAMON game and the next TAMON game. That is to say, to achieve an agreement, both players can play a series of TAMON game that is time constrained.

This simplification has two benefits:

1. The environment is stable during a TAMON game; therefore players can focus on the searching and improving of solutions without the need to deal with changing environmental parameters.

2. Existed time-dependent bidding strategies such as NDF [3] can be applied directly to a series of TAMON game without conflicting with "micro" level strategies used in the TAMON game.

This assumption also indicates that the goal of micro level strategies will be to find a solution given the constraints from "macro" level strategies. And macro level strategies are applied on a series of TAMON game.

### 3.3 The Information State

Each player (let  $b$  denotes buyer and  $s$  denotes seller) in a TAMON game is modeled as a 5-tuple  $I^a$  with a utility function  $U$ , a utility threshold  $U_{threshold}$ , a degree of information sharing  $D$ , a micro level strategy  $S$  and a mutually agreed period of time  $T$  to play the game:

$$I^a = \langle U^a, U_{threshold}^a, D^a, S^a, T \rangle, \text{ where } a \in \{b, s\}. \quad (1)$$

We define the utility function to be a sum of weighted contributions of  $N$  issues:

$$U = \sum_{i=1}^N w_i * u_i, \text{ where } u_i \in [0,1] \text{ and } \sum_{i=1}^N w_i = 1. \quad (2)$$

The utility contributed by each issue, however, is not necessarily depending only on the value of this specific issue, but also on the values of  $K$  other issues in the offer. The occurrences of this interdependency will make it inappropriate to be negotiated in an issue-by-issue way.

The micro level strategy  $S$  is defined by the algorithm and the algorithm parameters used for generating and accepting offers (detailed in section 5). In this paper, we will assume there is only one choice of algorithm, the BGA (discussed in the next sub-section), and  $D$  does not change during the series of TAMON game. Therefore the macro level strategies contain only functions determining the value of  $U_{threshold}$ . The search space bounded by  $U_{threshold}$  in a TAMON game is illustrated in Figure 2.

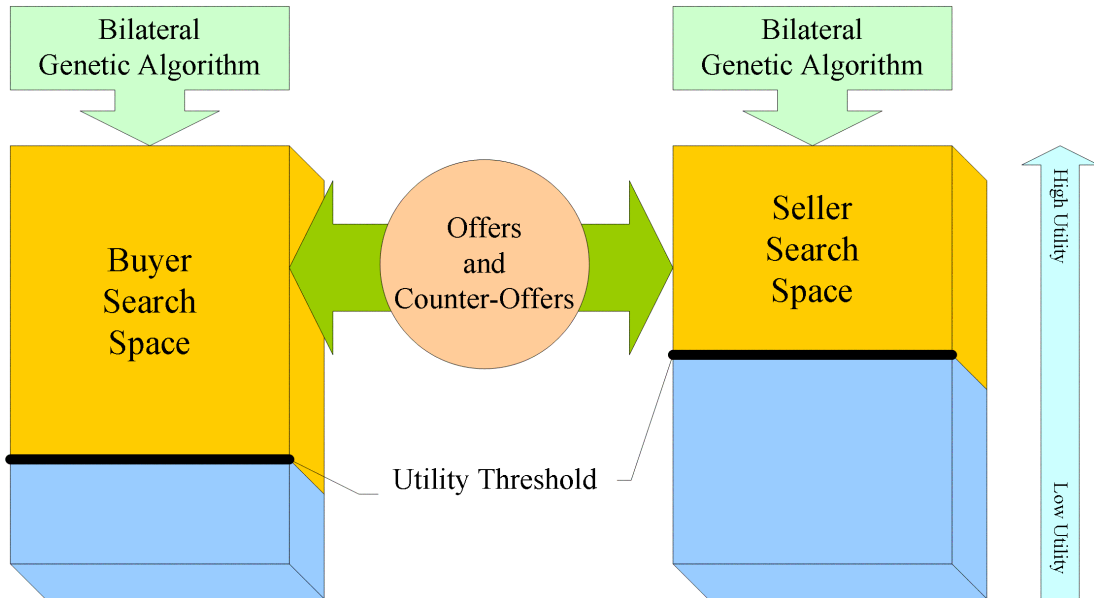


Figure 2. The Search Space in a TAMON Game

### 4. The Bilateral Genetic Algorithm

To search in the space of a TAMON game, we need a heuristic method that is computationally tractable; since the overall search space is too large that an exhaustive fashion of search is not possible. Considering a single-issue negotiation for delivery time in Figure 3, if both negotiators are using the same linear time-dependent macro level function for decreasing the  $U_{threshold}$ , their agreement can be found at the intersection of utility  $x$ . In this case, the offer generating process would be too simple that the micro level TAMON game can be set to propose only one offer in a turn and be played for only two turns (in the first turn, the buyer proposes and the seller accepts

or rejects, in the second turn the seller proposes and the buyer accepts or rejects). However, in a multi-issue search space, it is almost impossible to generate all the offers with utility  $x$  (while there are only two offers of utility  $x$  evaluated using  $f$  or  $g$  in Figure 3, there can be a huge number of offers of utility  $x$  in a multi-issue search space, especially when given nonlinear utility functions). Therefore, the searching algorithm can only propose some of them, and there is a great chance that the search algorithm will miss the right one. Suppose that a solution is guessed out at utility  $x'$  of time  $y''$  later, and then a mechanism will be required to back-search for the real optimal solution of utility  $x$ . The Bilateral Genetic Algorithm (BGA) is proposed to deal with this problem.

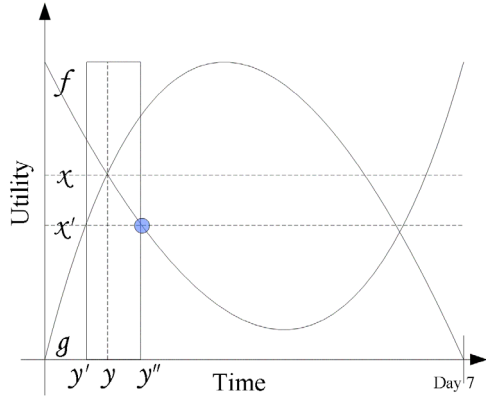


Figure 3. The Matching Point

#### 4.1 Overview of BGA

BGA is an algorithm for searching solutions to the TAMON problem. We apply the concepts of genetic algorithm (GA [8]) to the domain of offers. That is, we use a joint utility to express the fitness of an offer, and use an evolutionary approach to find out the best offer if possible. A very simple design of such an idea is to encode an offer into a single chromosome and let the fitness of it be the product (product is used instead of sum for fairness) of two utility values from the buyer and the seller. For example, let a chromosome  $\{100, 5, 2, 3\}$  (for readability, we use n-ary gene encoding) represents an offer with price of 100, quantity of 5 units, delivery time of 2 days and warranty period of 3 years. If the joint utility of this chromosome is high, the number of this individual will grow exponentially in the population, as stated in GA. And the population will eventually converge to the answers we want.

However, since the agents are self-interested in our scenario, each agent does not share information of his

utility with the other. Lacking the utility function from the other agent, an agent will not be able to determine the joint utility of a chromosome. Hence, both the buyer agent and the seller agent are not able to perform the selection using genetic operators on the population of offers. To overcome this problem, we propose an innovative algorithm BGA, which divides the population for evolution into a buyer side population and a seller side population, and uses two special genetic operators, B-selection and B-recombination, to handle the selection and recombination process in the presence of incomplete fitness function. Both agents must perform the genetic B-selection process and B-recombination process separately. The B-selected population will be proposed to the other player while the received population will be used to B-recombine with our B-selected population, as illustrated in Figure 4. The B-selection has a tunable sampling rate, which can select offers for proposing according to the  $D$  parameter. We will discuss the details of each operation in the following sub-sections, and then explain why it works.

Before we discuss the genetic operators used in BGA, we firstly define the parameters used in BGA:

Population Size:  $Z_{population}^a \in \mathbb{N}$ .

Crossover Rate:  $R^a = [0,1]$ .

Mutation Rate:  $M^a = [0,1]$ .

Chromosomes:  $C_i^a$ , where  $i \in \mathbb{N}$  and  $1 \leq i \leq Z_{population}^a$ .

Again:  $a \in \{b, s\}$ .

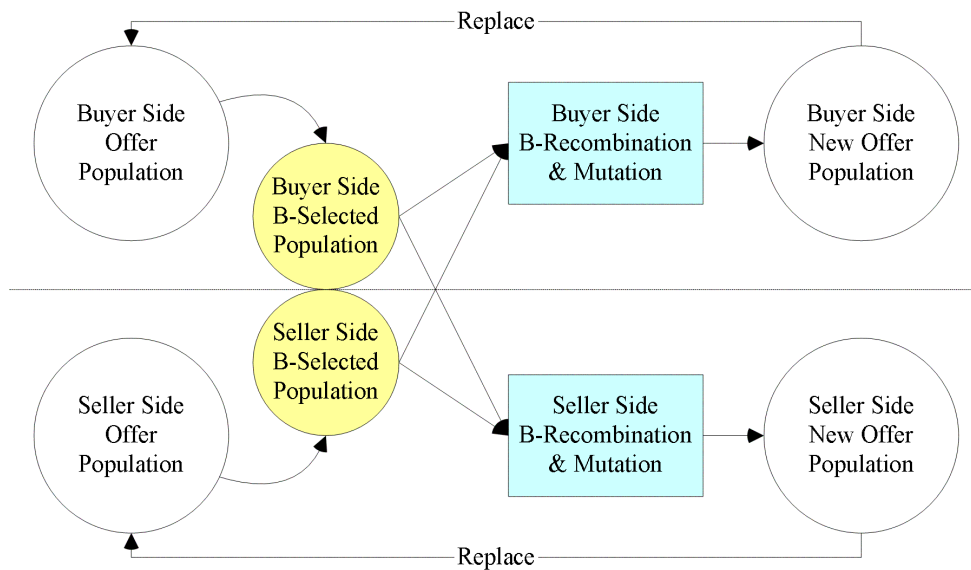


Figure 4. The BGA Evolution Process

#### 4.2 B-selection, B-recombination and B-mutation

The selection process of a GA will produce a new population with the distribution of chromosomes being proportional to the fitness of each chromosome from the old one. The one with higher fitness gets higher probability to be selected in the resulted population. B-selection does a little more: it will propose the population of chromosomes to the other agent, and the size of it will be affected by the degree of information sharing (parameter  $D$ ). The actual size of population proposed is:

$$Z_{offers} = \lceil Z_{population} \times D \rceil. \quad (3)$$

During the agent communications, duplicated offers can be represented using a vector format  $\langle \text{Offer, Number} \rangle$ . The B-selected population then becomes offers proposed to the other player in a tagged multi-offer format. The higher value in  $D$ , the richer information in the proposed offers. If  $Z_{offers}$  equals to 1, it becomes a normal alternating offer protocol that proposes only one offer a turn.

The B-recombination operator differs from the GA recombination in that it recombines chromosomes from the different-side B-selected populations. In other words, one of the parent chromosomes comes from the buyer-side B-selected population while the other from the seller-side B-selected population. Since the population B-selected by the other agent might have different size from ours, both the received population and our B-selected population are rescaled to half of our population size before they can be put together into a joined population of size  $Z_{population}$  (see Figure 5). The joined population is then ready for B-recombination. It should be noted that, each agent might have a joined population of different size, since their  $Z_{population}$  values might be different.

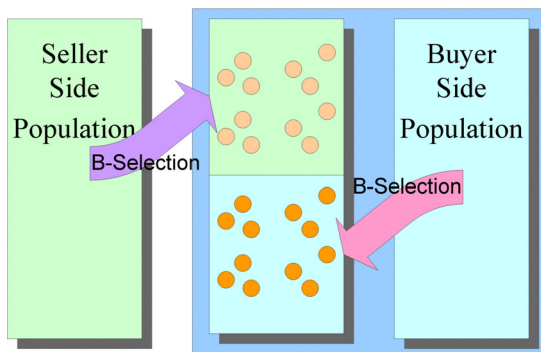


Figure 5. The Joined Population

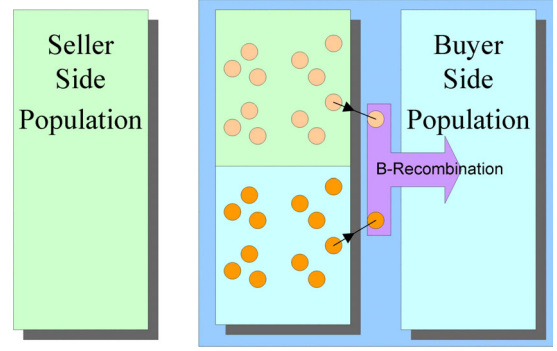


Figure 6. Performing B-recombination

B-recombination of two chromosomes produces chromosomes that represent offers applying different concession strategies. For example, recombining a buyer proposal  $\{100, 4, 2, 4\}$  with a seller proposal  $\{120, 3, 2, 2\}$  may produce a proposal  $\{100, 3, 2, 2\}$  denoting that the seller concedes at the price of value 100, and a proposal  $\{120, 4, 2, 4\}$  denoting that the seller concedes at quantity, delivery time and warranty period. Each agent performs B-recombination at his own memory space; therefore the crossover rates can be different at the two sides. After the B-mutation process, the newly generated population then replaces the old one (noted that the first generation is randomly generated). The B-mutation is similar to the GA mutation process except that it is performed separately at both sides.

We use uniform crossover in B-recombination, although it is considered to be maximally disruptive when the epistatic interactions are the nearest neighbors [9]. We believe it is required in our scenario, because it helps the B-recombination generate more possibilities of concessions assuming that the two negotiating agents may have different information spaces. For example, if the buyer can propose only one chromosome  $\{A, B, C\}$ , and the seller can propose only one chromosome  $\{D, E, F\}$ , uniform crossover makes it possible to generate a chromosome with  $\{A, E, C\}$  while one-point crossover cannot.

In the final algorithm, the B-selection is actually processed twice (Figure 7) by an agent in a turn; one for proposing offers, and one for generating half of population for next join. We need to separate these two populations since the proposed one is bounded by the utility threshold while the reserved one for next join is not. The detail of the algorithm is not explained in this paper. For interested readers, please check the web site: <http://homepage.ntu.edu.tw/~d85725004/BGA.html>.

```

GENERATE-OFFERS( $S_{pop}^r, D_{is}^r, R_{rec}^r, R_{mut}^r, b_f^r, b_u^r, \psi^r, \mu^r, \rho^r, \sigma^r, \theta^r, \vartheta^r, \chi^r$ )
1  $\chi^r \leftarrow \chi^r \cup \theta^r$ 
2 if ( $\theta^r = \phi$ ) or ( $\vartheta^r = \phi$ )
3   then  $\alpha \leftarrow \text{CREATE-POPULATION}(\rho^r, S_{pop}^r)$ 
4   else  $\alpha \leftarrow \text{B-MUTATE}(\text{B-RECOMBINE}(\text{RESCALE}(\theta^r, S_{pop}^r/2), \vartheta^r, R_{rec}^r), R_{mut}^r, \rho^r)$ 
5  $\gamma \leftarrow \text{ASSIGN-FITNESS}(\alpha, \psi^r, b_f^r)$ 
6  $\vartheta^r \leftarrow \text{B-SELECT}(\gamma, S_{pop}^r/2, 0)$ 
7  $\beta \leftarrow \text{B-SELECT}(\gamma, [S_{pop}^r \cdot D_{is}^r], b_u^r)$ 
8  $\theta^r \leftarrow \beta - \chi^r$ 
9  $\chi^r \leftarrow \chi^r \cup \theta^r$ 
10 return  $\theta^r$ 

GENERATE-FEEDBACK( $D_f^r, \theta^r, \mu^r, b_u^r, \psi^r$ )
1 if  $\sigma^r \neq \phi$ 
2   then  $b_u^r \leftarrow U(\sigma_1^r)$ 
3    $\psi^r \leftarrow \psi^r \cup \sigma^r$ 
4 if  $\theta^r \neq \phi$ 
5   then  $\gamma \leftarrow \text{AGREEMENTS}(\theta^r, \mu^r, b_u^r)$ 
6    $b_u^r \leftarrow \max\{(\gamma_j)_2 | 1 \leq j \leq \text{size-of}(\gamma)\}$ 
7    $\psi^r \leftarrow \psi^r \cup \gamma$ 
8    $\sigma^r \leftarrow \text{FEEDBACK}(\gamma, D_f^r)$ 
9 else  $\sigma^r \leftarrow \phi$ 
10 return  $\sigma^r$ 

```

Figure 7. The BGA algorithm

### 4.3 Preliminary Agreements

At the beginning of a BGA process, all chromosomes (i.e. offers) with utility higher than or equal to  $U_{threshold}$  are assigned a fitness value  $F_{threshold} = 0.99 * U_{threshold}$ . The coefficient 0.99 is designed to make the fitness value a little smaller than the  $U_{threshold}$ . Once a preliminary agreement (a proposed offer that is accepted) is reached, the associated chromosome will be given fitness equals to its real utility, which is higher than  $F_{threshold}$ . The number of this chromosome will then start to increase because of its high utility (further explained in the next sub-section). The discovery of a preliminary agreement also causes the value of  $U_{threshold}$  to climb up to the utility equaling to the one of the preliminary agreement. This climbing behavior ensures that the newly proposed offers will have a higher utility than preliminary agreements. It is the back-searching mechanism mentioned in the first paragraph of this section. In the view of macro level strategies, the value of  $U_{threshold}$  often goes down (decreasing) and does not go up, only when some preliminary agreement has been reached, it then makes sense to revert the  $U_{threshold}$  to the new high utility reached. This action is named a *utility threshold reverting*.

### 4.4 Implicit Parallelism in BGA

The main idea of BGA is to utilize the implicit parallelism of GA to explore all possible concession strategies and accumulate useful building blocks [7] at one

time. Implicit parallelism, named by Holland [8], is a property that:

*...Even though each generation we perform computation proportional to the size of the population, we get useful processing of something like  $n^3$  schemata in parallel with no special bookkeeping or memory other than the population itself...*

In BGA, only preliminary agreements contain useful building blocks that need to be accumulated in the later search. That is why we restore fitness of chromosomes to their actual utility value after they are found to be preliminary agreements (before that, all fitness values are lower than  $U_{threshold}$ ).

By B-recombining chromosomes from two populations of different sides, chromosomes representing various offer-improving possibilities are generated. The offer-improving behavior is a little bit like the similarity approximation in [4], since B-recombination try to generate an offer by recombining chromosomes from the two different populations and the resulted chromosomes will be similar to their parents. In fact, the BGA algorithm does a better job than [4] for locating offers similar to the opponent's. Because research [4] try to find similar offers by decreasing the utility distance, which is impossible if certain kinds of information about the opponent's utility function is unavailalbe. Their simulations work simply because there are too many assumptions (linear, conflicting, same value range and equal discrimination power over the reservation values) being placed on the opponent's utility function in their research. It makes the assumption of "incomplete information" quite weak.



B-recombination does not have a fixed-pie bias, since it does not assume a utility conceding is necessary when trying to generate an alternative. In fact, the BGA never decrease the utility of new offers, and try to increase the utility whenever possible. Once an offer acceptable to both sides is found, the utility threshold is increased, and the useful genetic information in the offer will then be accumulated in an evolutionary way.

To conclude, BGA can deal with the searching problem of a TAMON game in a computationally tractable way, and no specific assumptions are placed upon the utility function of the opponent.

## 5. Experimental Analysis

A negotiation case with mixed (including linear and nonlinear) utility function is prepared to test the capability of BGA. The utility functions for 5 issues in buying a car are listed in Table 1 (for simplicity, some of the functions are not showed in detail; mechanisms used to ensure the range of each utility value are omitted; for interested readers, complete source codes can be found at <http://homepage.ntu.edu.tw/~d85725004/BGA.html>). Two exceptions are added to override the default utility functions. They provide extra interdependency among these issues, and are the epistatic interactions as explained in [9].

**Table 1. Utility functions**

	Buyer	Seller
Price	$(200 - ((v*v)/200)) / 200$	$(v-20)/300$
Time	$\text{Cos}( (2*\text{Pi}*(v \bmod 7))/7 )$	$\text{Sin}( (2*\text{Pi}*(v \bmod 30))/30 )$
Type	{0.1, 0.2, 0.9, 0.5, 0.1}	$v/4$
Color	{0.3, 0.4, 0.2, 0.7, 0.9}	{0.5, 0.5, 1, 0.5, 0.5}
Option	$(10-v)/10$	$v/10$
Exceptions	If (Type = 1) and (Color = 2) then both issues get an utility value of 0.9.	If (Time = 77) and (Type = 1) then both issues get an utility value of 1.

Eighteen combinations of parameters (Table 2) are used to run the simulations. They are designed to answer the following questions:

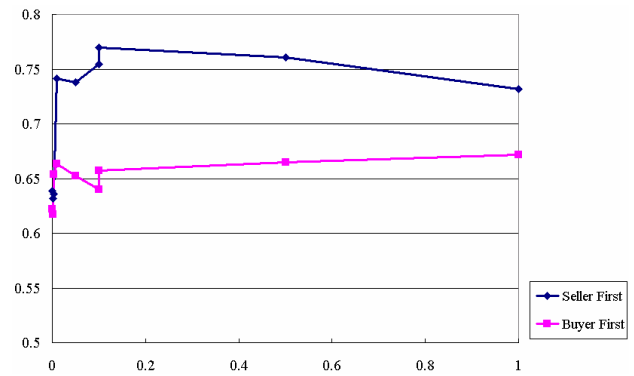
1. Should I be the first mover? (should I propose offers firstly?)
2. How do I propose? (how much information can be disclosed?)
3. How do I response? (how much information should be fed back to the opponent regarding his proposing?)

In question 2, the degree of information sharing determines the sampling rate for proposing offers in mind, and in question 3, the feedback rate of preliminary agreements determines how much information regarding one's evaluation on the opponent's offers will be fed back to him.

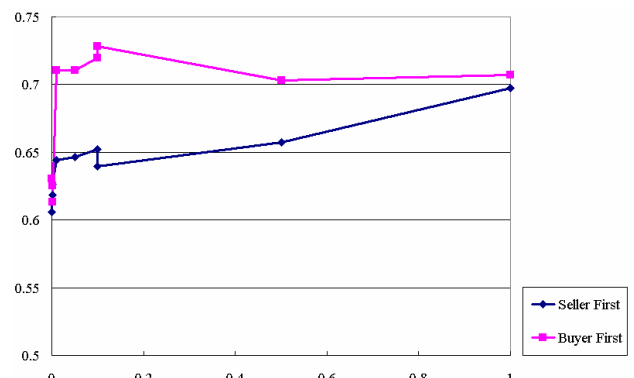
**Table 2. Experiment parameters**

Q	Parameter Name	Parameters
1	First Mover	seller first buyer first
2	Degree of Information Sharing (DoIS for short)	1 (all) 0.1 0.0025
3	Feedback of Preliminary Agreements (FoPA)	1 (all) 0.5 0.1

As expected, the first mover has disadvantage in the TAMON game. The explanation is intuitive: since the first mover shares information firstly, the opponent is then able to increase his utility threshold firstly when preliminary agreements are found in the first turn. Readers can compare Figure 8 and Figure 9 to find out the difference (noted that the X-axis in them represents the product of DoIS and FoPA, and Y-axis the utility).



**Figure 8. Buyer utility**



**Figure 9. Seller Utility**

Judging from the results showed in the following Figure 10 (FoPA = 1) and Figure 11 (DoIS = 1), we can conclude that the effects of information sharing have higher impact on the negotiation results than feedbacks. The reason lies at that the number of preliminary agreements is small comparing to the proposed offers. Since both negotiators tend to decrease the utility threshold slowly in the progress of negotiation, the number of preliminary agreements can not be very large. We also found that the degree of information sharing need

not to be very high for satisfactory negotiation results to be gained.

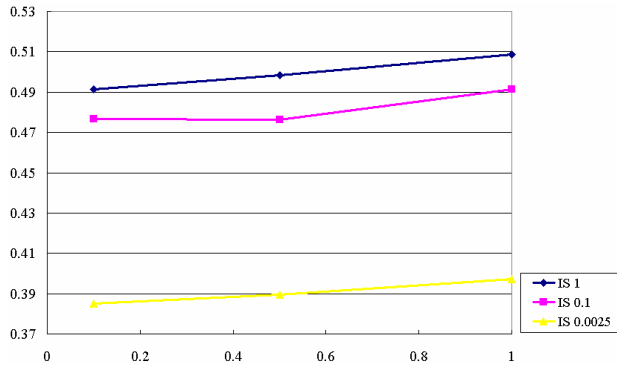


Figure 10. Effects of information sharing

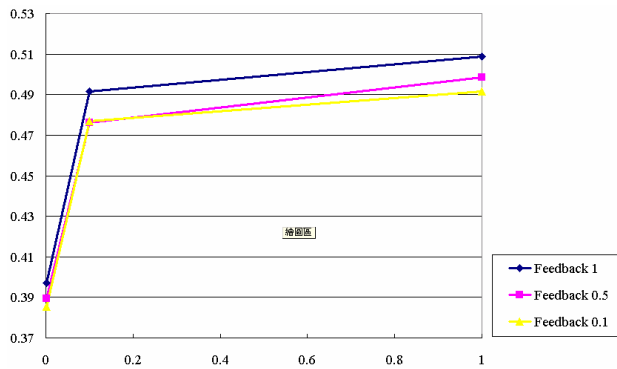


Figure 11. Effects of feedback

In Figure 12 (X-axis represents the buyer utility and Y-axis the seller utility, assuming FoPA = 1), the BGA simulation results are compared to GA simulations with same crossover rate (0.7) and mutation rate (0.02). Both population sizes are set to 400. In GA simulations, however, the algorithm has complete information of both negotiators' utility functions, and the product of buyer utility and seller utility is used as the fitness value. 1000 generations were run for the GA simulations while a period of 10 seconds was used in each micro level TAMON game simulation. We assume same bargaining power in the BGA simulations, therefore the utility threshold decreases at same speed (0.1 per micro level game) for both negotiators. Both GA and BGA simulations were run 30 times to get the average negotiation results in Table 3. The best result found by the brute force algorithm (testing all combinations) is also marked on Figure 12.

Table 3. Simulation results

	Buyer Utility * Seller Utility (Avg)	Ratio
Best	0.532946667	1
GA	0.520159512	0.976006689
BGA-1	0.508617115	0.954348993
BGA-0.5	0.491376085	0.921998608
BGA-0.0025	0.397237238	0.745360207

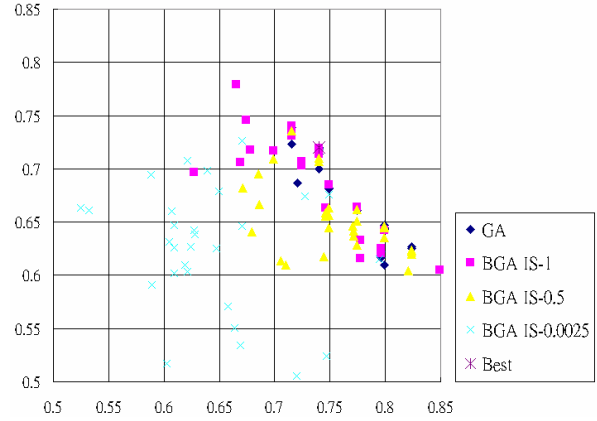


Figure 12. BGA simulations

The results in Table 3 show that BGA performs better when information is shared via the TAMON protocol. In the last case, where the degree of information sharing equals to 0.0025, only one offer was proposed in a turn (same as traditional negotiation model), and the ratio of average negotiation results drops dramatically. However, it is also found that by extending the TAMON game period, the results can be improved (see Figure 13, X-axis represents the degree of information sharing, and Y-axis the resulting utility product).

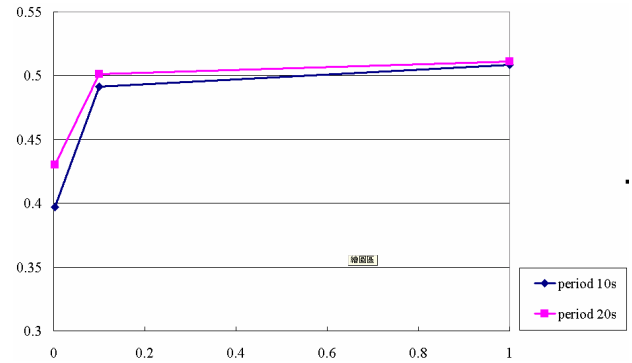


Figure 13. Effects of negotiation period

## 6. Conclusion and Future Work

We proposed the TAMON negotiation model and the BGA algorithm in this paper. The TAMON model defines a micro two-player negotiation game without time constraints, and allows more information to be exchanged in the negotiation game. The exclusion of time constraints is important for simplifying a TAMON game, since most search algorithms are not of real-time, and the search space tends to be static during the searching. It also permits a more complex search process to be conducted in a TAMON game, with a constraint that the process should be terminated after a period of time. By allowing more information to be disclosed in a tagged multi-offer format, the TAMON model incorporates the information sharing behavior into the negotiation model. We can say that the TAMON model provides a dimension for joint problem solving possibilities, yet it is still in the context of a non-cooperative game.

The BGA proposed in this paper utilizes the implicit parallelism of GA to search the solutions in a TAMON game. The elegance of the BGA is that it perfectly fits in the TAMON game since the population to be proposed is meaningfully transformed into the tagged multi-offer format, and with the degree of information sharing being taken into consideration. The algorithm is immune to the fixed-strategy limitation and the fixed-pie bias, and it is computationally tractable. It should be noted that it is not required that both players use the same algorithm in the negotiation. However, since the BGA can explore multiple strategies simultaneously, we are investigating that given the limitation of computational tractability, whether the BGA-like algorithm will be the only rational choice in a TAMON game.

We are developing a theory to measure the information disclosed in the tagged multi-offer protocol and to determine the precise effects caused by the information sharing in a multi-issue negotiation. A theory for guiding the choosing of BGA parameters, including the population size, the crossover rate and the mutation rate will also be addressed in the future work.

## References

- [1] Bazerman, M.H. and Carroll, J.S. "Negotiator cognition," *Research in Organizational Behavior* (9), B.M. Staw and L.L. Cummings (eds.), Greenwich, CT: JAI Press, 1987.
- [2] Chavez, A. and Maes, P. "Kasbah: An agent marketplace for buying and selling goods," 1st Intl Conf on Practical Appli. of Intelligent Agents Technology, 1996.
- [3] Faratin, P., Sierra, C. and Jennings, N.R. "Negotiation decision functions for autonomous agents," *International Journal of Robotics and Autonomous Systems*, 1998, (24:3-4), 159-182.
- [4] Faratin, P., Sierra, C. and Jennings, N.R. "Using similarity criteria to make trade-offs in automated negotiations," *Artificial Intelligence and Int. Journal of Autonomous Agents and Multi-Agent Systems*, 2003. (to appear, downloadable at <http://www.ecs.soton.ac.uk/~nrj/download-files/aij03-faratin.pdf>)
- [5] Foroughi, A. "Minimizing negotiation process losses with computerized negotiation support systems," *The Journal of Applied Business Research*, 1998, (14:4), 15-26.
- [6] Goh, K.Y., Teo, H.H., Wu, H. and Wei, K.K. "Computer-supported negotiations: an experimental study of bargaining in electronic commerce," in Proceedings of the 21st Annual International Conference on Information Systems, Brisbane, Australia, 2000, 104-116.
- [7] Goldberg, D.E. *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.
- [8] Holland, J.H. *Adaptation in natural and artificial systems*, The MIT Press, 1992.
- [9] Hordijk, W. and Manderick, B. "The usefulness of recombination," In F. Moran, A. Moreno, J.J. Merelo, and P. Chacon (eds.), *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life*, Springer-Verlag, 1995, 908-919.
- [10] Lomuscio, M.W. and Jennings, N.R. A classification scheme for negotiation in electronic commerce. In F. Dignum and C. Sierra, editors, *Agent-Mediated Electronic Commerce: A European Agentlink Perspective*, Springer-Verlag, 2001.
- [11] Maes, R.P. and Moukas, A.G. "Agents that buy and sell," *Communications of the ACM*, 1999, (42:3), 81-91.
- [12] Matwin, S., Szapiro, T., and Haigh, K. "Genetic algorithms approach to a negotiation support system," *IEEE Transactions on Systems, Man, and Cybernetics*, 1991, (21:1), 102-114.
- [13] Oliver, J.R. "A machine learning approach to automated negotiation and prospects for electronic commerce," *Journal of Management Information Systems*, 1997, (13:3), 83-112.
- [14] Raiffa, H. *The Art and Science of Negotiation*, Harvard University Press, Cambridge, USA, 1982.
- [15] Rubinstein, A. "Perfect equilibrium in a bargaining model," *Econometrica*, 1982, (50:1), 155-162.
- [16] Sandholm, T. "Agents in electronic commerce: component technologies for automated negotiation and coalition formation," *Autonomous Agents and Multi-Agent Systems*, 2000, (3:1), 73-96.
- [17] Sycara, K.P. "Negotiation planning: an AI approach," *European Journal of Operational Research*, 1990, (46), 216-234.
- [18] Thompson, L.L. and Hastie, R. "Social perception in negotiation," *Organizational Behavior and Human Decision Processes*, 1990, (47), 98-123.