Determining Successful Negotiation Strategies: The Evolution of Intelligent Agents

Jong Yih Kuo

Department of Computer Science and Information Engineering Fu Jen Catholic University, Taipei jykuo@csie.fju.edu.tw

Abstract

Due to the desire of almost all departments of business organizations to be interconnected and to make data accessible at any time and any place, more and more multi-agent systems are applied to business management. As numerous agents are roaming through the Internet, they compete for the limited resource to achieve their goal. In the end, some of them will succeed, while the others will fail. However, when agents are initially created, they have little knowledge and experience with relatively lower capability. They should also strive to adapt themselves to the changing environment. It is advantageous if they have the ability to learn and evolve. This paper addresses evolution of intelligent agents in virtual enterprises. Agent fitness and fuzzy multi-criteria decision-making approach are proposed as evolution mechanisms, and fuzzy soft goal is introduced to facilitate the evolution process. Genetic programming operators are employed to restructure agents in the proposed multi-agent evolution cycle. We conduct a series of experiments to determine the most successful strategies and to see how and when these strategies evolve depending on the context and negotiation stance of the agent's opponent.

1. Introduction

Doing business on the Internet is becoming more and more popular. The use the Internet to facilitate commerce among companies and customers brings forth many benefits, such as automated transactions, greater access to buyers and sellers, and dramatically reduced costs. The agent-based e-commerce has emerged and become a focus of the next generation of e-commerce. The intelligent agent act on behalf of customers to carry out delegated tasks automatically. They have demonstrated tremendous potential in conducting various e-commerce activities, such as comparison-shopping, auction, sales promotion, etc [1,2].

In order to solve a problem, an agent has to have certain skills and the ability to reason about these skills. We call the reasoning abilities as "mental" skills [3]. However, when agents are initially created, they have little knowledge and experience with relatively lower capability. They should also strive to adapt their negotiation strategies and tactics to the changing environment. It is advantageous if the agents have the ability to learn and evolve. Many issues are essential in agent evolution. Firstly, evolution of an agent is closely related with agent structure. Thus, a

The Second International Conference on Electronic Business Taipei, Taiwan, December 10-13, 2002 suitable agent structure is one of basic concerns in agent evolution. Secondly, agents should have their own mechanisms is advance evolution. Thirdly, in multi-agent system, evolution of individual agent is also related with many social concerns, such as coordination, negotiation, communication, etc. Finally, some tools can be used to evaluate the fitness of agents in the evolution procedures.. In this paper, we address multi-agent evolution for agents in e-commerce. Section 2 summaries our service-oriented negotiation model based on fuzzy theory and the BDI model. In Section 3, we adopt an evolutionary approach in which strategies and tactics correspond to the genetic material in a genetic algorithm. In Section 4, we present an empirical study showing the relative success of different strategies against different types of opponent in different environments. Section 5 contains our conclusion.

2. The Service-oriented Negotiation Model

This paper addresses evolution of intelligent agents about the mental skills. Obviously, there is no limit to what one would like to include under what we call mental skills. We agree that BDI model [4,5,6] provides a simple but powerful formalism for the representation, the specification and the analysis of the mental attributes of intelligent agent: belief, desire and intention

2.1 The BDI Model

In the BDI architecture an agent can be completely specified by the events that it can perceive, the actions it may perform, the beliefs it may hold, the goals it may adopt, and the plans that give rise to its intentions [7]. The figure 1 represents the relationships of BDI model.



Figure 1: The BDI model of Agent

A belief model describes the information about the environment and internal state that an agent of that class may hold, and the strategies and tactics it may perform. A goal model (desires) describes the goals that an agent may possibly adopt, and the events to which it can respond. It consists of a goal set which specifies the goal and event domain and one or more goal states – sets of ground goals – used to specify an agent's initial mental state. There are soft and rigid goals specified by the users. We use fuzzy logic to represent the goals [8].

A plan model (intensions) describes the plans that an agent may possibly employ to achieve its goals. A plan is a sequence of strategies through reasoning mechanism (mental skills of the agent). The strategy is the combination of tactics with various weights.

2.2 The Goal-driven Analysis

To model user the BDI model, we use GDUC (goal-driven use case) approach [9] to structure the goals hierarchy and to analyze the plans or strategies achieving these goals. The steps describe below.

- (1) Identify actors and user's goals to construct belief model: First, we must analyze the organization of enterprise or the environment of e-commerce to extract the basic knowledge for the agent. The knowledge can be build into a general common ontology. We also identify the users and their preferences to build the specific user-defined ontology. The ontology hierarchy can be stored into the knowledge-based of belief model.
- (2) Analyze goal hierarchy to build goal model: A faceted classification is proposed for identifying goals from domain descriptions and system requirements. Each goal can be classified under three facets we have identified: competence, view and content. The facet of competence is related to whether a goal is completely satisfied or only to a degree. A rigid goal describes a minimum requirement for a target system, which is required to be satisfied utterly. A soft goal describes a desirable property for a target system, and can be satisfied to a degree. The facet of view concerns whether a goal is actor-specific or system-specific. Actor-specific goals are objectives of an actor in using a system; meanwhile, system-specific goals are requirements on services that the system provides. We use the "use case" to structure goals hierarchy.
- (3) Analyze goal model to build plan model: According to the user's goal and use cases, we can construct the scenarios of use cases and the possible planes to achieve the goals. Then we also evaluate the degrees of satisfaction about the planes. The ability of context sensitivity and evolution help agent adopt the negotiation strategies to achieve user's goals.

2.3 Applying Fuzzy Theory to BDI Model

To model user goals, we apply GDUC to get a set of soft and rigid goals, a set of use cases, and a set of planes. For achieving these goals, agent must use particular strategies to change their mental states. We can continuously change the problem state to achieve the goal state. Thus we can apply the soft requirement [10] to formally represent the user goals.

A user goal, g, is specified by the properties of

agent's mental state-transition *<b, g, a>*, where *b* is the state before a plan, and *a* is the state after invoking the plan. A plan or strategy can thus be specified using a pair *<precondition, post-condition>*. The precondition and the post-condition describe properties that should be held by the state *b* and *a*. A rigid goal describes state properties that must be satisfied. The soft goal describes state properties that can be satisfied to a degree. We use Zadeh's test-score semantic [11] to represent the user goals. A basic idea underlies test-score semantics is that a proposition *p* in a natural language may be viewed as a collection of elastic constraints, $C_{I_n, \dots, n} C_k$, which restricts the values of a collection of variables $X = (X_{I_n}, \dots, X_n)$. In fuzzy logic, this is accomplished by representing p in the canonical form:

$$G \Longrightarrow R(P) IS A \tag{1}$$

in which *A* is a fuzzy predicate. The canonical form *G* implies that the possibility distribution of *R*(*P*) is equivalent of the membership function of *A*, namely $\Pi_{R(P)} = \mu_A$. For example, the agent helps a user to buy high quality item and can be represented using the canonical form below:

$$G \Rightarrow Quality(goods) \text{ IS HIGH}$$
 (2)

Where *HIGH* is a fuzzy predicate. The rigid goal is the specialization of the soft goal, which membership function of fuzzy predicate is 1.0.

2.4 The Negotiation Strategies

A negotiation strategy is the combination of tactics with various weights. A tactic generates a value for a single negotiation issue based upon a single criterion (e.g. time remaining, resource remaining).

As negotiation proceeds, the goals of agent may become relevant and the relative importance of existing criteria may vary. To reflect this fact, an agent has a strategy that varies the weights of the different tactics over time in response to various environmental. We extended the research [12,13] to proposed four types of tactics described below.

(1) Time-dependent tactics

These tactics model the fact that the agent is likely to concede more rapidly as the negotiation deadline approaches. The negotiation must have completed at the pre-established deadline (t_{max}) . The maximum price is P_r . When the deadline is nearly up, the price approaches the P_r . The function of tactics:

$$f_t = \alpha_t(t) P_r.$$
(3)
$$\alpha_t(t) = k_t + \left(1 - k_t\right) \left(\frac{t}{t_{\max}}\right)^{1/\beta t}$$

$$0 \leq \alpha_t(t) \leq l, \ \alpha_t(0) = k_t, \ \alpha_t(t_{max}) = l$$

 $0 \leq k_t \leq 1, 1/200 \leq \beta_t \leq 1000.$

(2) Resources-dependent tactics

These tactics generate offers depending on how a particular resource is being consumed; they become

progressively more conciliatory as he quantity of resource diminishes. Here, we use the bidder tactics. The equation is:

$$f_r = \alpha_r(t) P_r.$$

$$\alpha_r(t) = k_r + \left(1 - k_r\right) \left(\frac{c(t)}{|A|}\right)^{1/\beta r}$$

$$0 \le k_r \le 1, 1/200 \le \beta_r \le 1000.$$

$$(4)$$

c(t) is the number of web at $0 \sim t$, |A| is the number of active bidding web at $0 \sim t_{max}$.

(3) Prices-dependent tactics

Agent uses these tactics to maintain the goal of minimum price. Agent must get the biding prices of all active bidding webs. The equation is:

$$f_{p} = \omega(t) + \alpha_{p}(t)(P_{r} - \omega(t))$$

$$\alpha_{p}(t) = k_{p} + \left(1 - k_{p} \left(\frac{t}{t_{\max}}\right)^{1/\beta p} \right)$$

$$0.1 \leq k_{p} \leq 0.3, \ 1/200 \leq \beta_{p} \leq 0.5$$

$$\omega(t) = \frac{1}{|L(t)|} \left(\sum_{1 \leq i \leq |L(t)|} \frac{t - \sigma_{i}}{\eta_{i} - \sigma_{i}} v_{i}(t)\right)$$
(5)

|L(t)| is the number of active bidding webs at time t. The η_i represents the start time of the *i*th bidding web. The σ_i is the end time of the *i*th bidding web. The $\upsilon_i(t)$ represents the highest price of the *i*th bidding web at time *t*. (4) Desire-dependent tactics

Agent does the best to buy the high quality goods to achieve the user desired. The curve of the price will quickly approach the P_r . The equation is:

$$f_{d} = \omega(t) + \alpha_{d}(t)(P_{r} - \omega(t))$$

$$\alpha_{d}(t) = k_{d} + \left(1 - k_{d}\right) \left(\frac{t}{t_{\max}}\right)^{1/\beta d}$$

$$0.7 \leq k_{p} \leq 0.9, \ 1.67 \leq \beta_{d} \leq 1000$$
(6)

3. The Evolution of Intelligent Agent

Genetic algorithm (GA) operators are employed to restructure agents in the proposed multi-agent evolution cycle. How to encode a solution of the problem into a chromosome is a key issue for genetic algorithms. In Holland's work [14] encoding is carried out using binary strings. For many GA applications, the simple GA was difficult to apply directly because the binary string is not a natural coding. During the past ten years, various bon-string encoding techniques has been created for particular problems, for example, real number coding for constrained optimization problems and integer coding for combinatorial optimization problems. In our research, a real-coded GA uses floating-point numbers to represent genes [15].

3.1 Coding Schema

In order to find proper intelligent agent, the agent's negotiation strategies are coded and represent an individual. A strategy that is the combination of tactics with various weights will determine the bidding price at time *t*. We have three categories of tactics: time- dependent, resource-dependent, behavior- dependent. The equation of a strategy describes below:

$$S(t) = w_t f_t + w_r f_r + w_p f_p + w_d f_d$$
(7)

$$0 \le w_t, w_r, w_p, w_d \le 1 ,$$

$$w_t + w_r + w_p + w_d = 1$$

Each agent is represented as a string of fixed length. The bits of the string (the gene) represent the parameters of the agent's strategy.

$$G = (t_{max}, d, r, k_1, \beta_1, w_1, k_2, \beta_2, w_2, k_3, \beta_3, w_3, k_4, \beta_4, w_4)$$
(8)

3.2 Measuring a Strategy's Fitness

A fitness function is the survival arbiter for individuals. For finding the near-optimal intelligent agent, we propose the fuzzy multi-criteria decision- making (FMCDM) approach as the evolution mechanisms, and the fuzzy soft goals to facilitating the evolution process.

We may analyze the user's goals into a goal model. Each goal can have some criteria. The evaluation of soft goal is a satisfaction degree. The relationships between goals will exist conflicting and cooperative. Most of the existing approaches in multiple criteria making lack the aspect of an explicit modeling of relationships between goals. Carlsson and Fuller [16] advocated that much closer to MCDM in the real world than the traditional MCDM are the case with interdependent criteria. Our previous work on Criteria Trade-off Analysis (CTA) has been on the formulation of soft criteria based on Zadeh's canonical form in test-score semantics and an extension of the notion of soft condition [17]. The trade-off among soft goals is analyzed by identifying the relationships between goals. A compromise overall satisfaction degree can be obtained through the aggregation of individual goal based on the goals hierarchy. The steps describe below.

(1) To compute the relationship between goals: The c, c` are two soft goal, and a is the strategy, CF and CP denote the set of conflicting and cooperative pairs. AP denotes the set of all pairs. The conflicting and cooperative degree between two goals is defined as:

$$cf(c,c') = \frac{\sum_{(aiaj) \in Cf} \{|\mu(a) - \mu(aj)| + |\mu'(a) - \mu'(aj)|\}}{\sum_{(aiaj) \in Af} |\mu(a) - \mu(aj)| + |\mu'(a) - \mu'(aj)|\}}$$
(9)

$$P(c,c') = \frac{\sum_{(ai,aj) \in CP} (|\mu(ai) - \mu(aj)| + |\mu(ai) - \mu(aj)|)}{\sum_{(ai,aj) \in Ap} (|\mu(ai) - \mu(aj)| + |\mu(ai) - \mu(aj)|)}$$
(10)

0

- (2) To covert connections of the goals into DNF (Disjunctive Normal Form), and to establish a goals hierarchy: We assume that goals specified by users are connected by linguistic connectives in natural language. To take these connectives into account, we proposed the use of DNF to obtain a uniform representation of the goals. According to the conflicting and cooperative degrees, a goals hierarchy of *n* levels is defined as a tree. This tree is important in the sense that the ordering established through the hierarchy helps alleviate the associative problem inherited in fuzzy aggregation operator.
- (3) By using fuzzy aggregation operator to compute the strategy's fitness: An extended goals hierarchical aggregation structure is proposed to facilitate goals aggregation through the fuzzy and / or operator. The fitness can be obtained through the aggregation of satisfaction degrees based on the aggregation structure.

3.3 The Evolution Steps

All GAs use some form of mechanism to chose which individuals from the current population should go into the mating pool that forms the basis of the next population generation. A selection mechanism known to work well in such circumstances is Tournament Selection [18]. The crossover process exchanges genetic material between individuals. We randomly select two individuals from the population. Crossover points are then randomly chosen and sorted in ascending order. Then the genes between successive crossover points are alternately exchanged between the individuals, with a probability. Mutation process works by randomly selecting some of the genes present in the population in order to mutate.

The evolution of agent describes below.

(1) Initial population

A GA requires a population of potential solutions to be initialized at the beginning of the GA process. Here, we randomly generate some genes to create the initial population. We also generate some genes based on the agent's belief model.

(2) Selection Procedure

Selection procedure may create a new good population for the next generation based on either all parents and offspring or part of them [19,20] • . A sampling space is characterized by two factors: size and ingredient (parent or offspring). In regular sampling, there are several replacement strategies to replace old parents with offspring when new offspring are produced. As mentioned before, the population of next generation was formed by roulette wheel selection [19]. When selection performs on enlarged sampling space, both parents and offspring have the same chance of competing for survival. An evident advantage of this approach is that we can improve GA performance by increasing the crossover and mutation rates. We don't worry that the high rate will introduce too much random perturbation if selection is performed on enlarged sampling space.

A reproduction operation allows strings that with higher fitness values would have larger number of copies while the strings with lower fitness values have a relatively smaller number of copies or even none at all. This is an artificial version of natural selection (strings with higher fitness values will have more chances to survive). For example, suppose that N strings are generated, and the fitness value of the *ith* individual string is f_i (*i*=1, ..., N). Then, the probability of the ith individual string to be selected into the mating pool is

$$p_i = \frac{f_i}{\sum\limits_{i=1}^N f_i},\tag{11}$$

and the number of copies for the individual string is calculated by

 $n_i = N \cdot P_i \tag{12}$

This strategy emphasizes the survival-of-the-fitness aspects of the GA. The better strings receive more copies and go into the mating pool so that their desirable characters may be passed onto their offspring.

(3) Crossover

Crossover is a process to provide a mechanism for two high-fitness strings (parents) to produce two offspring by matching their desirable qualities through a random process [21]. The procedure of crossover is to select a pair of strings from the mating pool at random, then, an integer position k (called the crossover point) along the string is selected uniformly at random between 1 and (1-1), where 1 is the string length greater than 1. Finally, according to the probability of crossover, two new strings are generated by swapping all characters between position (k+1) and 1 inclusively. For example, consider two strings A and B of the population are mated for crossover

A=011|11001

B=100|10011

Suppose we obtain k=3 as indicated by the separator symbol "|". The resulting crossover yields two new strings as shown below

A'=01110011

B'=10011001

where A' and B' are the strings of the new generation. Although the crossover is done by random selection, it is not the same as a random search through the search space. Since it is based on the reproduction process, it is an effective means of exchanging information and combining portions of high-fitness solutions.

An agent is composed of some kinds of modules. Each module has a special feature for a particular plan. In multi-agent system, there are two phase of crossover: the inter-agent and intra-agent. With the inter-agent crossover, an agent exchanges a module with another agent. The intra-agent crossover exchanges some parameters or strategies between modules in the same agent.

(4) Mutation

Mutation is a process to provide an occasional random alteration of the value at a particular string

position [22]. In the case of binary string, this simply means changing the state of a bit from 1 to 0 and vise versa. A uniform mutation is first to produce a mask randomly, then change the selected string value in the position of mask where the bit value is "1". For example, consider the following selected string and generated mark:

$$A' = 0\ 1\ 1\ 1\ 0\ 0$$

M=100101

then, the 0-1 pattern of the string becomes as the following string,

A"=111100.

mask

Mutation occurs with a small probability in the GA to reflect the small rate of mutation existing in the real world. Mutation is needed because some digits at a particular position in all strings may be eliminated during the reproduction and crossover operations. Such a situation is impossible to be recovered by using only reproduction and crossover operations. To ensure that reproduction and crossover do not loose some potentially useful genetic materials (1's or 0's at particular locations), in mutation phase, some bits will be changed in all the strings according to the mutation rate (P_m) . In general, the mutation rate is less than 0.05. So the mutation plays a role as a safeguard in GA. It can help GA to avoid the possibility of mistaking a local optimum for a global optimum.

4. Experiments and Results

Over the last few years, the number of online auction houses has increased tremendously. To date there are more than 760 auction houses that conduct business online. Some examples of popular online auction house include eBay, Amazon, Yahoo!Auction, Priceline, Ubid, and FirstAuction. The types of auction that are conducted vary from site to site, but the most popular one are English, first-price sealed bid. In the English auction, the auctioneer begins with the lowest acceptable price and bidders are free to raise their bids successively until there are no more offers to raise the bid. The winning bidder is the one with the highest bid [23].

4.1 The Simulation Environment

In this paper, a digit video is the target item. The P_r is its reservation price for the target item. The bidder is given a deadline by when it needs to obtain the item. There are five predefined auctions running in the environment. These auctions have a finite start time and duration generated randomly from a standard probability distribution. The start time and the end time vary from one auction to another. The auction starts with a predefined small starting value. The process is repeated until the reservation price is reached or until the end time for the auction is reached. At the start of each auction, a group of random bidders are generated to simulate other auction participants. These participants operate in a single auction and have the intention of buying the target item and possessing certain behavior. They maintain information about the item they wish to purchase, their private

valuation of the item (reservation price), the starting bid value and their bid increment. These values are generated randomly from a standard probability distribution. They start bidding at starting bid value; when making a counter offer, they add their bid increment to the current offer, and they stop bidding when they acquire the item or when their reservation price is reached.

4.2 The Strategy of the Bidder Agent

The bidder agent is allowed to bid in any of the auction at any time when the marketplace is active. The objective of the bidder agent is to participate across the multiple auctions, bid in the auctions and deliver the item to its consumer in a manner that is consistent with their preferences. The bidder agent utilizes the available information to make its bidding decision; this includes the use's reservation price, the time it has left to acquire the item, the current offer of each individual auction, and its set of tactics and strategies. The output of the bidding decision is the auction the agent should bid in and the recommended bid value that it should bid in that auction. The agent's overall behavior is the amalgamation of those strategies proposed in this paper, weighted by their relative importance to the user. Mapping this to an auction environment, the bidder agent needs to decide the new bid value based on the current offer price. Let t be the current universal time across all auctions, where $t \in \tau$, and τ is a set of finite time intervals. Let t_{max} be the time by when the agent must obtain the good (i.e. $t_{max} \ge t \ge 0$), and let A be the list of all the auctions that will be active before time t_{max} . At any time t, there is a set of active auctions L(t) where $L(t) \subset A$. Since each auction has a different start and end time, the bidder agent needs to build an active auction list to keep track of all the auction that are currently active in the marketplace. The agent identifies all the active auctions and gathers relevant information about them. It then calculates the maximum bid it is willing to make at the current time using the agent's strategy. Based on the value of the current maximum bid, the agent selects the potential auctions in which it can bid and calculates what it should bid at this time in each such auction. The auction and corresponding bid with the highest expected utility is selected from the potential auctions as the target auction. Finally, the agent bids in the target auction.

4.3 Experimental Evaluation

Our experiments were run in an environment with 10 agents of the first generation, $t_{max} = 100$, 5 English auctions running concurrently, and for each auction, there are 10 participants. If the reservation price of the agent is reached or until the end time for the all auction is reached, we can get the bidding price, the bidding time, and the quality of the target item for each agent. We apply the fuzzy decision making approach to compute the fitness value of agent. Then we apply the evolution approach to generate the next generation agents. In this particular experiment, the mutation rate is 0.02. The simulations stop when the

population is stable (95% of the individuals have the same fitness) or the number of iterations is bigger than a predetermined maximum (200 in our case).

5. Conclusion

In this paper, we present a new approach for evolving intelligent agents in e-commerce. A goal-driven approach can construct the user's soft and rigid goals based on fuzzy set theory. The proposed BDI model represents the mental skills of the intelligent agent, including belief, desire, intension, and strategy. A FMCDM approach is applied to evaluate the agent's strategy. Agent fitness and life cycle are proposed to facilitate and control the process of agent evolution. We construct multi-agent evolution cycle, which includes states of restructuring, selection, and growing. We conduct a series of experiments to determine the most successful strategies and to see how and when these strategies evolve depending on the context and negotiation stance of the agent's opponent. Finally, we have demonstrated the usefulness of agents employing a cocktail of tactics - both for different negotiation issues and, in combination, for a single issue.

6. Acknowledgements

This paper is supported by national science council under grants NSC-90-2213-E- 030-014.

References

- R.H. Guttman, and P. Maes. Agent-mediated negotiation for retail electronic commerce. In Agent-mediated Electronic Commerce: First International Workshop on Agent Mediated Electronic Trading, Berlin: Springer, pp. 70-90, 1999.
- [2] T.H. Wang, S.U. Guan, and S.H. Ong. An agent-based auction service for electronic commerce. In Proceedings of International ICSC Symposium on Interactive and Collaborative Computing, Australia, 2000.
- [3] D. C-H Han and N. Parameswaran. Multi Agent Problem Solving with Mental States. IEEE. pp. 288-292. 1994.
- [4] P. Cohen and H.J. Levesque. Intention is Choice with Commitment. Artificial Intelligence, 42(3): pp. 213-216, 1990.
- [5] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. Reasoning About Knowledge. MIT Press, 1995.
- [6] A. Rao and M. Georgeff. Modeling rational agents within a BDI architecture. J. Allen, R. Fikes, and E. Sandewall (editors). KR-91. Morgan Kaufmann, 1991.
- [7] M. Barbuceanu and M.S. Fox. COOL: A language for

describing coordination in multi-agent systems. In Proceedings of the International Conference on Multi-Agent Systems, ICMAS-95, San Francisco, CA, 1995.

- [8] J. Lee and J.Y. Kuo. New approach to requirements trade-off analysis for complex systems. IEEE Transactions on Knowledge and Data Engineering.10 (4): 551-562. 1998.
- [9] J. Lee, N.L. Xue and J.Y. Kuo. February 2001. Structuring Requirement Specifications with Goals. Information and Software Technology, Vol. 43, pp. 121-135.
- [10] J.Lee, J. Y. Kuo and A. Liu. Modeling User Requirements with a Front-end Analysis. International Journal on Artificial Intelligence Tools. 6(3): 313-324, Sep. 1997.
- [11] L.A. Zadeh. Test-score Semantics as a Basis for a Computational Approach to the Representation of Meaning. Literacy Linguistic Computing, 1:24, 1986.
- [12] P. Anthony, W. Hall, V. Dung, N.R. Jennings. Autonomous for Participating in Multiple Online Auctions. Proceedings IJCAI Workshop on E-Business and the Intelligent Web, Seattle, WA, pp. 54-64 ,2001.
- [13] N. Matos, C. Sierra and N. R. Jennings. Determining successful negotiation strategies: an evolutionary approach. Proceedings 3rd Int. Conf. on Multi-Agent Systems (ICMAS-98), Paris, France, pp. 182-189, 1998.
- [14] J. Backer, "Adaptive Selection Methods for Genetic Algorithms", Proceedings ISCGA, pp. 100-111. 1994.
- [15] M. Michalalewicz, Genetic Algorithms + Data Structure = Evolution Program, Springer-Verlag, Berlin, 1992.
- [16] C. Carlsson and R. Fuller. Interdependence in fuzzy multiple objective programming. Fuzzy Sets and System, 65:19-29, 1994.
- [17]J. Lee and J.Y. Kuo. Fuzzy decision making through trade-off analysis between criteria. Information Sciences, 107(1-4): 107-126. 1998.
- [18] D. Goldberg. Genetic Algorithms in Search Optimization and Matching Learning. Addison-Wesley. 1989.
- [19] Baker, J.B., "Adaptive Selection Methods for Genetic Algorithms", Proceedings of the First International Conference on Genetic Algorithms, pp.101~111, 1985.
- [20] D. Goldberg. Genetic Algorithms in Search. Optimization and Matching Learning. Addison- Wesley. 1989.
- [21] Chen, T.Y. and Chen, C.J., "Improvements of Simple Genetic Algorithm in Structural Design", International Journal for Numerical Methods in Engineering, Vol.40, pp.1323-1334, 1997.
- [22] Z. Michalewicz, T. Logan, and S. Swaminath. Evolutionary Operators for Continuous Convex Parameter Spaces", In Proceedings of the 3 rd annual Conference on Evolutionary Programming. World Scientific, pp. 84-97, 1994.
- [23] R. P. McAfee and J. McMillan. Bidding rings. American Economic Review, Vol. 82 No. 3. pp. 579-599. 1992.