Evolving Best-Response Strategies for Market-Driven Agents Using Aggregative Fitness GA

Kwang Mong Sim, *Senior Member, IEEE*, and Bo An

***Abstract*—There are very few existing works that adopt genetic algorithms (GAs) for evolving the most successful strategies for different negotiation situations. Furthermore, these works did not explicitly model the influence of market dynamics. The contribution of this work is developing bargaining agents that can *both*: 1) react to different market situations by adjusting their amounts of concessions and 2) evolve their best-response strategies for different market situations and constraints using an aggregative fitness GA (AFGA). While many existing negotiation agents only optimize utilities, the AFGA in this work is used to evolve best-response strategies of negotiation agents that optimize their utilities, success rates, and negotiation speed in different market situations. Given different constraints and preferences of agents in optimizing utilities, success rates, and negotiation speed, different best-response strategies can be evolved using the AFGA. A testbed consisting of both: 1) market-driven agents (MDAs)—negotiation agents that make adjustable amounts of concessions taking into account market rivalry, outside options, and time preferences and 2) GA-MDAs—MDAs augmented with an AFGA, was implemented. Empirical results show that GA-MDAs achieved higher utilities, higher success rates, and faster negotiation speed than MDAs in a wide variety of market situations.**

***Index Terms*—Automated negotiation, bargaining, genetic algorithms (GAs), negotiation agents.**

I. Introduction

AUTOMATED negotiation [29] among software agents is becoming increasingly important because automated interactions between agents can occur in many different contexts, and research on engineering e-negotiation agents [15], [24] has received a great deal of attention in recent years (see [15] and [24] for a survey). Even though there are many existing negotiation agents for e-commerce (e.g., [10]) and grid resource management [35], [38], [39], the negotiation strategies of agents in these systems seldom consider the dynamics of the market. In highly dynamic environments, such as a computational grid, it is essential to take market dynamics into consideration because providers can make resources/services available to and disconnect from a grid, and consumers can enter and withdraw requests, perhaps at machine speed in both cases. Furthermore, to operate successfully in open environments (e.g., a grid computing environment), bargaining agents must be capable of evolving their strategies to adapt to prevailing circumstances and constraints. Bargaining agents negotiating in less favorable market situations (e.g., facing more competition) and constraints (e.g., having limited time to negotiate and acquire resources) are expected to behave differently from those negotiating in favorable markets and perhaps without any deadline constraints. To adequately address real-world negotiation problems involving large agent populations (e.g., negotiation for resource allocation in a computational grid), bargaining agents need to be designed to both: 1) deal with a wide variety of bargaining (and market) situations and 2) be highly adaptable to learn and evolve to cope with dynamically changing circumstances.

To this end, the impetus of this work is to devise a mechanism using genetic algorithms (GAs) for learning and evolving the most effective bargaining strategies of Sim's market-driven agents (MDAs) [33], [36], [37], [41]–[43] (Section II) in different market situations and constraints. It was noted in [11] that GA is an alternative to the standard game-theoretic models for generating optimal solutions in a bargaining problem, particularly in practical situations involving large agent populations. By considering both market rivalry and outside options, it was proven in [34] and [37] that MDAs make prudent compromises (i.e., they do not make either excessive or inadequate concessions) in different market situations. Unlike some of the learning negotiation agents (e.g. [25]), MDAs in their present forms are not designed with capabilities that can enhance their performance by evolving their strategies. Nevertheless, unlike MDAs, many of the existing negotiation agents (e.g., [3], [4], [10], [21], and [25], just to name a few) do not take into consideration the influence of market factors such as market rivalry and outside options. To this end, this work will not compete with the existing works in negotiation agents, but rather it supplements and complements existing literature by developing bargaining agents that can both: 1) react to different market situations by adjusting its amounts of concessions and 2) evolve their best-response strategies for different market situations and constraints using an aggregative fitness GA (AFGA) (Section III).

Whereas many existing works only adopt utility as the performance measure for evaluating negotiation, in this work: 1) utility; 2) success rate; and 3) negotiation speed (measured in number of rounds needed to reach an agreement) are used (see Section IV). Although designing negotiation agents that only optimize utility (e.g., negotiating for the lowest possible price) may be sufficient for generic e-commerce applications, in some applications (e.g., grid resource management), negotiation

Manuscript received June 9, 2008; revised November 16, 2008. Current version published April 15, 2009. This work was supported by the DASAN Project (Gwangju Institute of Science and Technology (GIST) International Faculty Start-up Fund) from the Gwangju Institute of Science and Technology, South Korea. This paper was recommended by Associate Editor S. Leonid.

K. M. Sim is with the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea (e-mail: prof_sim_2002@yahoo.com).

B. An is with the Computer Science Department, University of Massachusetts, Amherst, MA 01002 USA (e-mail: ban@cs.umass.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2009.2014880

agents should also be designed to make tradeoff decisions (in addition to optimizing utility) so that they are more likely to acquire resources more rapidly and perhaps with more certainty. For example, in some application domains such as a grid computing environment, failure to obtain the necessary computing resources before a deadline may have an impact on the job executions of some applications and any delay incurred on waiting for a resource assignment is perceived as an overhead [40]. Hence, an AFGA (Section III) is used to evolve the strategies of MDAs for optimizing the three performance measures mentioned earlier in different market situations and time constraints.

A series of experiments (see Section IV) was carried out to: 1) determine the most successful strategies of MDAs for different market situations and time constraints and 2) compare the performance of MDAs [33], [36], [37], [41]–[43] against GA-MDAs (i.e., MDAs that are programmed to evolve the strategies using the AFGA). Empirical results (Section IV) show that GA-MDAs achieved higher expected utilities, higher success rates in reaching deals, and higher average negotiation speed than MDAs. While Section V compares the strengths and weaknesses of GA-MDAs with related systems, Section VI summarizes the contributions of this work.

II. Market-Driven Agents

An MDA is an e-negotiation agent that determines the appropriate amount of concession using three negotiation decision functions (NDFs): *time* (T), *competition* (C), and *opportunity* (O) [33], [36], [37], [41], [42]. It makes concession by narrowing the difference k_t between its proposal and the counter-proposal of its opponent in a negotiation round t. To determine the amount of concession, an MDA uses the T, C, and O functions to determine the difference k_{t+1} in the proposal and counter-proposal in the next round $t+1$

$$k_{t+1} = f\Big[O\Big(n_t^B, v_t^{B\to S_j}, \langle w_t^{S_j\to B}\rangle\Big), C\Big(m_t^B, n_t^B\Big), T(t,\tau,\lambda)\Big]k_t. \quad (1)$$

In an abstract market-driven model [33], a linear combination of all three factors is used to determine the overall concession. For the purpose of experimentation, following [41], this work used the product of the T, C, and O functions for determining overall concession.

Time function: In a bilateral negotiation, an MDA's decision is generally only influenced by time. $T(t,\tau,\lambda)$ models the process of an agent making a series of concession to narrow its difference(s) with other parties as time passes given as follows [33], [36], [37], [41], [42]: $T(t,\tau,\lambda) = 1-(t/\tau)^{\lambda}$, where t is the current trading time, τ is the deadline, and λ is an MDA's time preference. While deadline puts negotiators under pressure (see [18]), an MDA has different time preferences (e.g., negotiators with different time preferences may adopt different concession rates with respect to time) (see [28, pp. 32–33]). For instance, an agent may prefer to concede less rapidly in the early rounds of negotiation and more rapidly as its deadline approaches. In an MDA, the concession rate is determined with respect to time preference ($0<\lambda<\infty$) and deadline τ. With different values of τ, an MDA should adopt different strategies for making concessions by selecting different values of λ (details are given below). With infinitely many values of λ, there are infinitely many possible strategies in making concession with respect to remaining trading time. However, they can be classified as follows [37]:

1) *Linear*: $\lambda = 1$ and $k_{t+1} = [T(t,\tau,\lambda)]k_t = [1-(t/\tau)]k_t$. At any round t, an MDA makes a *constant* rate of concession $\triangle_t = k_t - k_{t+1}$. At the deadline $t=\tau$, $k_\tau = [1-(\tau-1/\tau)]k_{\tau-1}$ and $k_{\tau+1} = [1-(\tau/\tau)]k_\tau$. Hence, $\triangle_\tau = k_\tau - k_{\tau+1} = k_\tau$ (an MDA expects and attempts to narrow the difference completely at the deadline).
2) *Conciliatory*: $k_{t+1} = [1-(t/\tau)^{\lambda}]k_t$, where $0<\lambda<1$. An MDA makes larger concessions in the early trading rounds and smaller concessions at the later stage.
3) *Conservative*: $k_{t+1} = [1-(t/\tau)^{\lambda}]k_t$, where $1<\lambda<\infty$. An MDA makes smaller concessions in early rounds and larger concessions in later rounds.

Opportunity function: In a multilateral negotiation, having outside options may give a negotiator more bargaining "power." However, negotiations may still break down if the proposals between two negotiators are too far apart. The opportunity function

$$O\Big(n_t^B, v_t^{B\to S_j}, \Big\langle w_t^{S_j\to B}\Big\rangle\Big) = 1 - \prod_{j=1}^{n_t^B} \frac{v_t^{B\to S_j} - w_t^{S_j\to B}}{v_t^{B\to S_j} - c^B}$$

determines the amount of concession based on: 1) trading alternatives (number of trading parties n_t^B) and 2) differences in utilities ($v_t^{B\to S_j}$) generated by the proposal of an MDA and the counter-proposal(s) of its trading party (parties) ($\langle w_t^{S_j\to B}\rangle = \{w_t^{S_1\to B}, w_t^{S_2\to B}, \ldots, w_t^{S_{n_t^B}\to B}\}$) [33], [36], [37], [41], [42]. c^B is the worst possible utility for agent B. When negotiation ends in a conflict (i.e., B fails to reach an agreement with its trading parties), B obtains its worst outcome c^B. Whereas $v_t^{B\to S_j} - w_t^{S_j\to B}$ measures the cost of accepting a trading party's (S_j's) last offer (i.e., difference between the (counter-)proposals of B and S_j), $v_t^{B\to S_j} - c^B$ measures the cost of provoking a conflict. Details of deriving the O function are given in [37, p. 714].

Competition function: MDAs are designed for multilateral negotiations, and rivalry in an e-market is inherent. The amount of competition of an MDA is determined by the probability that it is not being considered as the most preferred trading party. The competition function $C(m_t^B, n_t^B)$ determines the probability that an agent B is ranked as the *most* preferred trading party by at least one other agent at round t [33], [36], [37], [41], [42]. If B has $m_t^B - 1$ competitors, and n_t^B trading parties, then $C(m_t^B, n_t^B) = 1 - [(m_t^B - 1)/m_t^B]^{n_t^B}$. Details of deriving the C function are given in [33], [36], [37], [41], and [42].

The general idea of the opportunity and competition functions is as follows. In favorable markets (i.e., when they have stronger bargaining positions), MDAs concede less. In unfavorable markets (i.e., when they have weaker bargaining positions), MDAs concede more. Some examples of the works that have followed up on the work of MDAs by considering the opportunity and/or competition functions include [12].

Sim [34], [37] has proven that agents adopting a market-driven strategy negotiate optimally by making *minimally sufficient concession* in different market conditions (see [37, Lemmas 4.1 and 4.2, pp. 718–719]). A concession is *minimally sufficient* if it achieves the *highest possible* utility for an MDA while maintaining a minimum probability of reaching a consensus (see [37, Definition 4.2, p. 718]). Even though conceding more increases the probability of reaching a consensus, it is inefficient because an MDA "wastes" some of its utility [37, p. 718]. However, if an MDA concedes too little, it runs the risk of losing a deal. Hence, MDAs are designed to avoid making excessive concessions in favorable markets and inadequate concessions in unfavorable markets. In a given market situation, an MDA is designed to attain the highest possible utility while maintaining a reasonable probability of reaching a consensus. With respect to opportunity and competition, Sim [37] has shown that the amount of concession made by an MDA is minimally sufficient (see [37, Lemmas 4.1 and 4.2, pp. 718–719]).

Problem Definition: In response to different bargaining deadlines, an agent should adopt different strategies (by selecting different values of λ) in making concessions with respect to time. With a longer deadline, an agent may find it advantageous to adopt a more conservative strategy since it has more time for negotiation. A conciliatory strategy may be more appropriate if an agent is facing a very tight deadline (e.g., it is coerced to acquire a resource rapidly). Sim has proven in [33] that with a longer (respectively, shorter) deadline τ, agents adopting conservative (respectively, conciliatory) strategies achieve higher (respectively, lower) utilities, but face higher (respectively, lower) risks of losing deals to competing agents. However, [33] has *only* provided mathematical analysis for showing that for larger τ (respectively, smaller τ), an agent should adopt a larger λ (respectively, smaller λ), corresponding to conservative (respectively, conciliatory) strategies. Finding an exact (or approximate) value of λ that would optimize both the utility and success rate of a market-driven bargaining agent under different τ is an open problem, and addressing this issue is the main impetus of this work. This work attempts to answer the question: "How should an MDA select the appropriate value of λ for its time function that determines the changing rate of $\triangle$ with respect to time, such that its utility is optimized with a reasonable probability that a consensus is reached?". With infinitely many values of λ, there are infinitely many time-dependent strategies for an MDA, and finding a solution to this optimization problem is difficult. One of the possible approaches for addressing this problem is to use a heuristic method such as GA to search the population of strategies for a best-response strategy. For different deadlines and market situations, the problem in this work is finding a best-response strategy that would optimize: 1) an MDA's utility; 2) its success rate; and 3) its negotiation speed. Please see Section III-B for a detailed problem formulation. Section III presents an AFGA to evolve a best-response strategy to optimize an MDA's utility, success rate, and negotiation speed for different deadlines and in different market situations.

Negotiation protocol: Negotiation proceeds in a series of rounds as follows. At round $t = 0$, the e-market opens. At any round, at most one agent enters the market randomly. Trading begins when there are at least two agents of the opposite type (i.e., at least one buyer and one seller). In the first round of trading, an agent proposes a deal from its space of possible deals that includes the most desirable (initial) price (IP), the least desirable (reserve) price (RP), and those prices in between IP and RP). Typically an agent proposes its most preferred deal initially. Adopting Rubinstein's alternating offers protocol [32, p. 100], a pair of buyer and seller agents negotiates by making proposals in alternate rounds. Many buyer–seller pairs can negotiate deals simultaneously. If no agreement is reached, negotiation proceeds to another round. At every round, an agent determines its amount of concession using the time, opportunity, and competition functions (see Section II). Negotiation between two agents terminates: 1) when an agreement is reached or 2) with a conflict when one of the two agents' deadline is reached.

e-Market structure: In summary, the e-market can have multiple buyers and multiple sellers. There can be a varying number of buyers and sellers in the e-market because agents can randomly enter or leave the market at different times. A buyer (respectively, seller) agent leaves the e-market when its deadline is reached or when it has reached an agreement with a seller (respectively, buyer) agent. While in the e-market, a buyer (respectively, seller) agent negotiates with all seller (respectively, buyer) agents in the e-market. Since the number of buyers and sellers in the e-market can vary, an agent's bargaining position at each negotiation round may also change. Based on its bargaining position, a buyer (respectively, seller) generates its proposal to each seller (respectively, buyer) using the opportunity, competition, and time functions that take into consideration the number of buyers and sellers in the market, the difference between the buyer's (respectively, seller's) proposal and the proposal of each seller (respectively, buyer), and the remaining trading time.

III. Aggregative Fitness GA

Evolutionary algorithms (such as GAs) are appropriate mechanisms for solving multiobjective optimization problems because they deal simultaneously with a set of possible solutions (represented as a population) that allows an entire set of Pareto optimal solutions to be found in a single run of the algorithm rather than performing a series of separate runs [5]. This section introduces an AFGA to learn and evolve the best-response strategies for different negotiation situations. An AFGA generates a sequence of populations as the outcome of a search method, modeled using a selection mechanism, crossover (recombining existing genetic materials in new ways) and mutation (introducing new genetic materials by random modifications). In this work, the individuals of the population are negotiating agents, and their genetic materials are the parameters of the market-driven negotiation strategies. Two populations representing buyer negotiation agents and seller negotiation agents are used. Both the buyer and seller populations coevolve simultaneously and dynamically by learning successful parameter settings for their respective survival, and the fitness of an individual in one population is based on direct competition with individuals from the other population [30]. While Section III-A describes the chromosome structure consisting of parameters that affect an MDA, details of the aggregative fitness of these negotiation strategies are given in Section III-B, and AFGA's search algorithm together with the GA operations are detailed in Section III-C.

A. Chromosome Structure

In determining an appropriate representation of the genes, the following is considered. Recall that k_{t+1} determines an MDA's amount of concession in the next round $(t+1)$ (see Section II), and the value of k_{t+1} is determined by

$$\left\langle v_t^{B\to S_j}, w_t^{S_j\to B}, n_t^B, m_t^B, t, \tau, \lambda \right\rangle$$

where $v_t^{B\to S_j}$, $w_t^{S_j\to B}$, n_t^B, m_t^B, and t are determined by the negotiation situation at round t. $v_t^{B\to S_j}$ is the utility generated by the proposal of agent B, $w_t^{S_j\to B}$ is the utility generated by the proposal of its trading party S_j, n_t^B is the number of trading parties, and m_t^B is the number of competitors.

Hence, only λ is used to determine the strategy of an MDA, where different values of λ represent different attitudes toward deadline (τ). To evolve the best-response negotiation strategies using AFGA, a society of negotiation agents with different strategies is created as the individuals of a population. The genetic materials of these individuals consist of the parameters (i.e., λ and τ) that affect the *MDA* strategies. Each agent is represented as a fixed-length string, and more specifically, the bits of a string (a chromosome) have the following structure:

1) λ: real. The value of λ determines an agent's strategy. Accordingly, genetic operations only take place on the bits of a chromosome that represent the value of λ.
2) τ: integer. The maximum time an agent can negotiate.
3) f: real. The fitness value.

B. Fitness Functions

An MDA's fitness value indicates how well it performs with the λ value that it adopts in comparison to others in the same population. Similar to basic evolutionary ideas, fitness determines an MDA's chance of surviving to the next population generation. With a higher fitness, an MDA is more likely to be selected for reproduction. To compute an agent's fitness, a round-robin tournament is played in which each buyer MDA negotiates with each seller MDA. Each MDA is assigned a fitness score that determines the performance of its strategy. The fitness value of a chromosome (corresponding to a strategy) is determined by the negotiation results as the chromosome negotiates with other chromosomes. Since the score of a negotiation strategy is determined by three objectives: utility (U), success rate (S), and negotiation speed (T), for this multiobjective problem, every solution has three fitness values (one for each objective). For example, for a negotiation result $X=(X_U, X_S, X_T)$, where X_U, X_S, and X_T are the negotiation results of all the three objectives of the negotiation result X, respectively, the function $f_I(X_I)$ is the performance measure I of the result X_I, in which $I \in \{U,S,T\}$. To normalize the three performance measures of a negotiation result to the same range ($f_I(X_I): X_I \to [0,1]$ here), the performance measure I of the result X_I, i.e., $f_I(X_I)$, is defined as

$$f_U(X_U) = \frac{X_U - \min(X_U)}{\max(X_U) - \min(X_U)} \tag{2a}$$

$$f_S(X_S) = \frac{X_S - \min(X_S)}{\max(X_S) - \min(X_S)} \tag{2b}$$

$$f_T(X_T) = \frac{\max(X_T) - X_T}{\max(X_T) - \min(X_T)} \tag{2c}$$

where $\max(X_I)$ and $\min(X_I)$ are, respectively, the maximum and minimum values of the performance measure I.

Hence, the fitness value $f(X)$ for the multiobjective optimization problem in this work can be expressed as follows:

$$\max_X \; f(X) = f(f_U(X_U), f_S(X_S), f_T(X_T)). \tag{3}$$

In defining $f(X)$, two issues need to be considered: 1) MDAs' constraints on multiple objectives. In this work, the cut-set technique in fuzzy mathematics [19] is used to describe these constraints. For an objective $I \in \{U,S,T\}$, each constraint can be described as $c_I \in [0,1]$; and 2) MDAs' preferences for placing the importance of the three objectives, which are represented as their priorities. In this work, the priority of an objective $I \in \{U,S,T\}$ is represented as $w_I \in [0,1]$. Both the constraints and preferences of MDAs for the three objectives for a negotiation result X are used to define the function $f(X)$, given as follows:

$$f(X) = \frac{\sum_{I\in\{U,S,T\}} (w_I \diamond \mu_I(f_I(X_I)))}{3} \tag{4}$$

where $\mu_I(f_I(X_I))$, $I \in \{U,S,T\}$, is defined as

$$\mu_I(f_I(X_I)) = \begin{cases} f_I(X_I), & \text{if } f_I(X_I) > c_I \\ 0, & \text{otherwise} \end{cases}$$

and operator $\diamond : [0,1]\times[0,1] \to [0,1]$, called a *priority operator*, satisfies [9, p. 50]

1) $\forall a_1, a_2, a_2' \in [0,1], a_2 \le a_2' \Longrightarrow a_1 \diamond a_2 \le a_1 \diamond a_2'$.
2) $\forall a_1, a_1', a_2 \in [0,1], a_1 \le a_1' \Longrightarrow a_1 \diamond a_2 \ge a_1' \diamond a_2$.
3) $\forall a \in [0,1], 1 \diamond a = a$.
4) $\forall a \in [0,1], 0 \diamond a = 0$.

In this work, the operator $\diamond : [0,1]\times[0,1] \to [0,1]$ is given by[1]

$$a_1 \diamond a_2 = (a_2 - 1) \times a_1 + 1. \tag{5}$$

Example: Let the priority set be $w_U = 0.9$, $w_S = 0.3$, and $w_t = 0.3$. Assume that $\mu_U(f_U(X_U)) = 0.8$, $\mu_S(f_S(X_S)) = 0.4$, and $\mu_T(f_T(X_T)) = 0.7$. Using (4) and (5), the fitness value $f(X)$ is given by $f(X) = ((0.9 \diamond 0.8) + (0.3 \diamond 0.4) + (0.3 \diamond 0.7))/3 = ((0.8-1)\times 0.9 + 1 + (0.4-1)\times 0.3 + 1 + (0.7-1)\times 0.3 + 1))/3 = 2.56$.

The value of the constraint $c_I \in [0,1]$ of objective I acts as a threshold for calculating the fitness value of a negotiation outcome. A larger value of c_I indicates that an MDA has

[1]The proof for the definition of the priority operator $\diamond$ is omitted due to lack of space.

rigorous requirement on objective I. For example, a combination $C = \langle c_U = 0, c_S = 0, c_T = 0\rangle$ models the situation that an MDA has no constraint on the multiple objectives. In contrast, a combination $C = \langle c_U = 0.8, c_S = 0.8, c_T = 0.8\rangle$ indicates that an MDA has rigorous requirement on the three objectives.

As agent designers may have different preferences for different objectives, it seems intuitive to place "weights" on the individual fitness values taking into account the constraints of the multiple objectives using the priority operator $\diamond$, and aggregating individual fitness using the weights to produce a single fitness value for every negotiation outcome, thus allowing the AFGA to rank all the negotiation outcomes. The priority operator $\diamond$ is used to express agents' different preferences on the three objectives. For instance, if an MDA has sufficient time for negotiation (e.g., 70 rounds—it was found by experimental tuning that the upper limit of deadlines is 70) and higher chance to reach an agreement, then it may place more emphasis on optimizing its utility, and less emphasis on negotiation speed and success rate. In contrast, when an MDA has shorter deadline for negotiation (e.g., 15 rounds—it was found by experimental tuning that the lower limit of deadlines is 15) and lower chance to reach an agreement, it may place more emphasis on negotiation speed and success rate.

The set of priority values of all the objectives can be elicited according to the relative importance of a subset of the three objectives. The value of w_I may be classified as follows:

1) $w_I = 0$ represents that an MDA does not consider the performance measure I.
2) $0 < w_I < 0.3$ represents that an MDA places low emphasis on the performance measure I.
3) $0.3 \leq w_I < 0.7$ represents that an MDA places moderate emphasis on the performance measure I.
4) $0.7 \leq w_I \leq 1$ represents that an MDA places high emphasis on the performance measure I.

For example, a combination $W = \langle w_U = 0.9, w_S = 0.3, w_T = 0.3\rangle$ indicates that an MDA places very high priority in optimizing its utility but at the same time, hopes to complete negotiation in a short time and maintains a certain level of possibility of reaching an agreement. Another combination $W = \langle w_U = 1, w_S = 1, w_T = 1\rangle$ indicates that an MDA places equal emphasis on all the three performance measures.

C. Genetic Algorithm

Details of the AFGA (see Fig. 1) are given as follows:

1) *Generation of the first population (Step 1).* In this step, the initial population is created by randomly generating chromosomes representing (both buyer and seller) MDAs with different values of λ.
2) *Fitness calculation (Step 2).* When a new population is generated, each chromosome representing a buyer (respectively, seller) MDA first negotiates with other chromosomes representing seller (respectively, buyer) MDAs, and the negotiation result obtained by each chromosome is used to determine its fitness using $f(X)$ (see Section III-B). The evolution of the strategies of individuals in the population representing buyer (respectively, seller) MDAs affects the strategies of individuals in the population representing seller (respectively, buyer) MDAs. For instance, as a buyer MDA adapts and evolves its strategy, the amounts of concessions it made to other seller MDAs would also cause other seller MDAs to revise and adapt their own strategies because other seller MDAs are also evolving their strategies as they negotiate with the buyer MDAs.

Step 0) Set the generation number t equal to zero: $t = 0$.
Step 1) Generate the initial population P_0.
Step 2) Calculate the fitness value of each individual in P_t.
Step 3) Scan the population P_t and save the one with the highest fitness: $Best_{P_t} = Best(P_t)$, and let $P'_{t+1} = (P_t - Best_{P_t})$.
Step 4) Implement Tournament Selection: P'_{t+1} = TS(P'_{t+1}).
Step 5) Implement Crossover: P'_{t+1}= Crossover (P'_{t+1}).
Step 6) Implement Mutation: P'_{t+1}= Mutation (P'_{t+1}).
Step 7) Let $P_{t+1} = P'_{t+1} \cup Best_{P_t}$
Step 8) IF Exit_Criteria=TRUE
  Stop
 ELSE
  Set $t = t + 1$
  Go to Step 2
 END IF.

Fig. 1. Algorithm for getting the best-response negotiation strategy.

3) *Elitist selection [1], [7] (Step 3).* The chromosome with the highest fitness Best_{P_t} in the current population P_t is selected and included as part of the new population P_{t+1} in the next round $t + 1$. Selection, crossover, and mutation are applied to the other $n - 1$ individuals (let them be P'_{t+1} here) in P_t.
4) *Tournament selection process (Step 4).* Tournament selection is used in the AFGA for selecting individuals from P_t for inclusion in the mating pool. Through tournament selection, k individuals are randomly selected from P_t. The individual with the highest fitness among the selected k individuals is placed in the mating pool. This process is repeated $n - 1$ times, where n is the size of the population. k is called the tournament size (in this work, k is 2) and it determines the degree to which the best individuals are favored. The tournament selection process generates $n - 1$ new chromosomes (that forms P'_{t+1}) for applying the crossover and mutation operations. Tournament selection is adopted in the AFGA in this work because using tournament selection, it is more likely that the mating pool will comprise a more diverse range of chromosomes (i.e., negotiation agents in this work). This is essential for facilitating a more thorough exploration of the search space, especially in the early stages.
5) *Crossover process (Step 5).* Crossover between chromosomes is carried out as follows. Two individuals from P'_{t+1} are randomly selected. The crossover operation is only performed on the bits representing the value of λ. Crossover points are randomly selected and sorted in ascending order. The chromosomes between successive crossover points are alternately exchanged between the individuals, with a

probability Pc. Through experimental tuning, a value of $Pc = 0.6$ is adopted.

6) *Mutation process (Step 6)*. Mutation is carried out as follows. Like crossover, it is only performed on the bits representing the value of λ. Some of the chromosomes in P'_{t+1} are randomly selected for mutation. For the selected chromosomes, a random value chosen from the domain of the chromosome is used to replace the value of selected bits representing the value of λ. The remaining chromosomes have a probability $Pm(t)$ of undergoing mutation where $Pm(t)$ is defined following [2] and is given as follows:

$$Pm(t) = \left(2 + \frac{l-2}{T-1}t\right)^{-1} \quad (6)$$

where l is the length of the string used to encode agents' strategies and T is the maximum number of iterations that the AFGA may run. From (6), it can be seen that: 1) at the start of the AFGA, the mutation rate is relatively high, e.g., when $t = 0$ (the initial population), $Pm(t) = 1/2$; 2) the mutation rate finishes with an optimal mutation rate [27] $Pm(t) = 1/n$ (that is proportional to l) when $t = T - 1$; and 3) $Pm(t)$ decreases with the increase of t since $d(Pm(t))/dt = -(l-2)/(T-1)(2 + (l-2)/(T-1)t)^{-2} < 0$, which is helpful with respect to the convergence reliability and velocity of a GA.

After applying selection, crossover, and mutation, the population in the next generation (P_{t+1}) is composed of the fittest individuals from the population P_t in the previous round together with $n - 1$ newly created individuals P'_{t+1}.

7) *Stopping criterion (Step 8)*. The algorithm stops when the population is stable (e.g., 95% of the individuals have the same highest fitness) or the number of iterations reaches a predetermined maximum (e.g., 100).

In summary, AFGA is applied to evolve an agent's best-response strategy in different negotiation environments (see Table III, for a summary of the different negotiation environments in terms of market density, market type, and deadline). In each negotiation environment, when the AFGA terminates, the chromosome in the final population that has the highest fitness is extracted. The value of λ of this chromosome, which has the highest fitness, is the best-response strategy for that negotiation environment.

For each given market situation, the λ value of the chromosome with the highest fitness will be adopted by the GA-MDA when it negotiates with other MDAs (adopting randomly selected value of λ) in the virtual e-market during the simulation.

IV. Evaluation and Experimentation

A. TestBed

To evaluate the performance of GA-MDAs (i.e., enhanced MDAs [33], [36], [37], [41]–[43] that are programmed to evolve the best-response strategies for different market situations and deadline constraints using an AFGA), a testbed is built. The testbed consists of: 1) a virtual e-market; 2) a society of negotiation agents comprising both MDAs and GA-MDAs; and 3) a controller (manager) agent. While the design of an MDA is described in Section II, a GA-MDA is an MDA that is augmented with an AFGA for evolving the best-response strategies in different market situations. The controller agent generates negotiation agents (both MDAs and GA-MDAs), randomly determines their parameters (e.g., their roles as either buyers or sellers, IP, RP, negotiation strategies, and deadlines), and simulates the entrance of agents to the virtual e-market. Agentsin the e-market adopt the following negotiation protocol in Section II.

TABLE I
Input Data Sources

Input Data	Possible Values		
Market Type	*Favorable*	*Balanced*	*unfavorable*
P_{Buyer}	< 0.5	0.5	> 0.5
P_{Buyer}: *Probability* of an agent *being a buyer*			
Buyer-seller Ratio	{1:2, 1:5, 1:10}	1:1	{10:1, 5:1, 2:1}
Market Density	*Sparse*	*Moderate*	*Dense*
P_{gen}	0.25	0.5	1
P_{gen}: *Probability* of *generating an agent* per round			
Deadline	*Short*	*Moderate*	*Long*
T_{max}	$18-25$	$35-45$	$60-70$

Using the testbed and adopting the negotiation protocol described in Section II, a series of experiments was carried out to compare the performance of GA-MDAs with that of the best-performing, the average-performing, and the worst-performing MDAs in a very wide variety of test environments. In the experiments, both GA-MDAs and MDAs were subjected to different market densities, different market types, and different deadlines (Table I).

B. Experimental Settings

All the three input parameters in Table I are randomly generated following a uniform distribution. Both market type and market density, respectively, depend on the probability of the agent being a buyer (or a seller) and the probability of generating an agent in each round. From a buyer agent's perspective, for a favorable (respectively, an unfavorable) market, an agent enters a market with lower (respectively, higher) probability of being a buyer agent and higher (respectively, lower) probability of being a seller. The lifespan of an agent in the e-market, i.e., its deadline, is randomly selected from $[15, 70]$. The range of $[15, 70]$ for deadline is adopted based on experimental tuning and agents' behaviors. In current experimental setting, it was found that: 1) for very short deadline (<15), very few agents could complete deals and 2) for deadlines >70, there was little or no difference in the performance of agents. Hence, for the purpose of experimentation, a deadline between the range of 18–25 (respectively, 35–45 and 60–70) is considered as short (respectively, moderate and long).

To enable GA-MDAs to evolve the best-response negotiation strategies for different negotiation environments in a *wide* strategy space, the search strategy space (the scope of λ) of GA-MDAs is $[0.001, 100]$ because through experimental

TABLE II
PERFORMANCE MEASURE

Success Rate	$R_{success} = N_{success}/N_{total}$
Expected Utility	$U_{expected} = U_{success} \times R_{success} + U_{fail} \times (1 - R_{success}) = U_{success} \times R_{success}$
Average Negotiation Speed	$R_{time} = \sum_{i=1}^{N_{total}} T^i_{end}/N_{total}$
N_{total}	*Total number of agents*
$N_{success}$	*No. of agents that reached consensus*
$U_{success}$	*Average utility of agents that reached consensus*
$U_{fail} = 0$	*Average utility of agents that didn't reach consensus*
T^i_{end}	*The time spent in negotiation by the agent i*

TABLE III
THE BEST-RESPONSE STRATEGIES OF GA-MDAS IN DIFFERENT MARKETS
$\left(c_U = 0, c_S = 0, c_T = 0, w_U = 0.9, w_S = 0.3, w_T = 0.3\right)$

Market Density	Market Type	Deadline		
		Long	Moderate	Short
Sparse	Favorable	1.60	1.41	1.23
	Balanced	1.45	1.19	0.89
	Unfavorable	1.08	0.87	0.57
Moderate	Favorable	3.90	2.12	1.66
	Balanced	2.76	1.63	1.09
	Unfavorable	1.42	1.11	0.63
Dense	Favorable	8.14	4.53	2.14
	Balanced	5.46	3.48	2.16
	Unfavorable	3.54	2.61	1.24

tuning, it was found that when $\lambda > 100$ (respectively, $\lambda < 0.001$), agents made little (respectively, large) compromises at the beginning of negotiation and there was almost no difference in performance of agents. After exploring different search spaces (e.g., $[0.001, 100]$, $[0.0005, 100]$, $[0.001, 300]$) of λ in different negotiation environments, it was found that the values of λ for GA-MDAs in different negotiation environments converged to the range $[0.1, 10]$ (see the values in Table III). To compare the performance of GA-MDAs and MDAs in the same strategy space, the values of λ used by MDAs in the experiments were generated randomly following a uniform distribution from the range $[0.1, 10]$. Following the guidelines prescribed in [34, Propositions 5 and 6], for long deadlines $(60 - 70)$ (see Table I), an MDA selects from the range $[>1, 10]$ (i.e., the class of conservative strategies in the range of $[0.1, 10]$), and for short deadlines $(18-25)$, it selects from the range $[0.1, <1]$ (i.e., the class of conciliatory strategies in the range of $[0.1, 10]$). Sim has proven in [34, Proposition 5, respectively, Proposition 6] that with a longer (respectively, shorter) deadline τ, agents adopting conservative (respectively, conciliatory) strategies achieve higher (respectively, lower) utilities, but face higher (respectively, lower) risks of losing deals to competing agents.

C. Performance Measure

Expected utility, success rate, and average negotiation speed are used as performance measures in the experiments (Table II). It was pointed out in [15] and [42] that other than optimizing agents' utility, enhancing the success rate is also an important evaluation criterion for designing negotiation agents. Since negotiation outcomes of each agent are uncertain (i.e., there are two possibilities: eventually reaching a consensus or not reaching a consensus), it seems more prudent to use expected utility (rather than average utility) as a performance measure because it takes into consideration the probability distribution over the two different outcomes [42]. Average negotiation speed is a measure of the average number of negotiation rounds needed to reach an agreement.

D. Results

An extensive amount of stochastic simulations was carried out for all the combinations of market density (dense, moderate, sparse), market type (favorable, almost balanced, unfavorable) and deadline (short, moderate, long). A total of 27 ($3 \times 3 \times 3$) combinations of the input parameters representing different negotiation environments were used. In addition, there are different combinations of constraints on multiple objectives and $3 \times 3 \times 3 = 27$ combinations of priority values for each negotiation environment. For a specific experimental environment, we tried different scenarios (more than 100) and obtained the average values for all performance measures. Even though experiments were carried out for all the situations subject to the constraint set, evaluation function, market density, market type, and deadline, due to space limitation, only *some* representative results are presented in this section. For the empirical results presented in this section, the combination of constraints $C = \langle c_U = 0, c_S = 0, c_T = 0\rangle$ is used in the experiments. Two sets of empirical results are presented in this section: 1) the best-response negotiation strategies evolved and obtained from the AFGA (Section IV-E) and 2) empirical results that compare the performance of GA-MDAs with MDAs (Section IV-F–H). Whereas Table III summarizes the best-response negotiation strategies for different market situations evolved using the AFGA, the empirical results that compare the performance of GA-MDAs with MDAs are plotted and shown in Figs. 2–19. In each of the graphs in Figs. 2–19, there are four curves: "GA-MDA," "Best-MDA," "MDA," and "Worst-MDA," respectively, representing the performance of GA-MDA, the best-performing MDA, the average-performing MDA, and the worst-performing MDA for the three performance measures: utility, success rate, and negotiation speed.

E. Evolving Best-Response Strategies

Sim [34] has proven that by adopting a larger value of λ, an MDA is more likely to achieve higher utilities, but at the same time, it may face higher risks of losing deals and perhaps more likely to reach agreements later. Hence, despite the variations in input parameters, any variation in the value of λ will also result in varying all the three performance measures in different negotiation situations. Since there are 27 combinations for the input parameters and 27 combinations of constraints on multiple objectives (see Section IV-D), there are $27 \times 27 = 639$ combinations of negotiation situations in which best-response strategies need to be evolved. Table III reports the best-response strategies of GA-MDAs for different market densities, market types, and deadlines for the priority combination $W = \langle w_U = 0.9, w_S = 0.3, w_T = 0.3\rangle$ (i.e., a GA-MDA places very strong emphasis on optimizing its utility, but hopes to complete negotiation in a reasonably short time and maintains a certain level of success rate in reaching an agreement). Similar experiments were also carried out for other representative combinations of the priority combinations, e.g., for $W = \langle w_U = 0.3, w_S = 0.9, w_T = 0.3\rangle$ and $W = \langle w_U = 0.3, w_S = 0.3, w_T = 0.9\rangle$. With $W = \langle w_U = 0.3, w_S = 0.9, w_T = 0.3\rangle$, the GA-MDA places very strong emphasis on reaching agreements successfully, but hopes to complete negotiation in a reasonably short time and maintains a certain level of utility. With $W = \langle w_U = 0.3, w_S = 0.3, w_T = 0.9\rangle$, the GA-MDA places very strong emphasis on completing negotiation rapidly, but hopes to achieve a certain level of utility and maintains a reasonably good chance of successfully reaching agreements. For $W = \langle w_U = 0.3, w_S = 0.9, w_T = 0.3\rangle$ and $W = \langle w_U = 0.3, w_S = 0.3, w_T = 0.9\rangle$, empirical results similar to those for $W = \langle w_U = 0.9, w_S = 0.3, w_T = 0.3\rangle$ were obtained. However, space limitation precludes the results for $W = \langle w_U = 0.3, w_S = 0.9, w_T = 0.3\rangle$ and $W = \langle w_U = 0.3, w_S = 0.3, w_T = 0.9\rangle$ from being included here.

Table III shows the best-response values of λ subjected to different market densities, market types, and deadlines. From Table III, it can be observed that:

1) Given the same market density and market type, but with shorter deadlines, GA-MDAs make more concessions by adopting a smaller value of λ. This observation corresponds to the intuition that with shorter deadlines, an agent faces higher risk of not reaching an agreement if it makes too small amounts of concessions. Failing to reach an agreement is not only an agent's worst outcome, but it will also lower the agent's success rate. Consequently, GA-MDAs react to more stringent time constraints by making larger amounts of concessions (i.e., adopting smaller values of λ).
2) Given the same market density and deadline, but in an unfavorable market, a GA-MDA makes more concessions by adopting a smaller value of λ. In an unfavorable market, a GA-MDA's bargaining position is weak, and it faces higher risk of not reaching an agreement if it makes too small amounts of concessions.
3) Given the same market type and deadline, but with lower market density, a GA-MDA makes more concessions by adopting a smaller value of λ. With low market densities, GA-MDAs have lower chance of reaching an agreement. To increase their probabilities of reaching an agreement, GA-MDAs make more concessions by adopting a smaller value of λ.

It is noted that each value of λ in Table III has two decimal places, and hence, the maximum error of representing the values of λ is lower than 0.01. Similarly, other values of λ can also be obtained for other different negotiation situations and fitness functions using the proposed AFGA. In the series of experiments that compares the performance of GA-MDAs with MDAs (see Section IV-F–H), GA-MDAs were programmed to

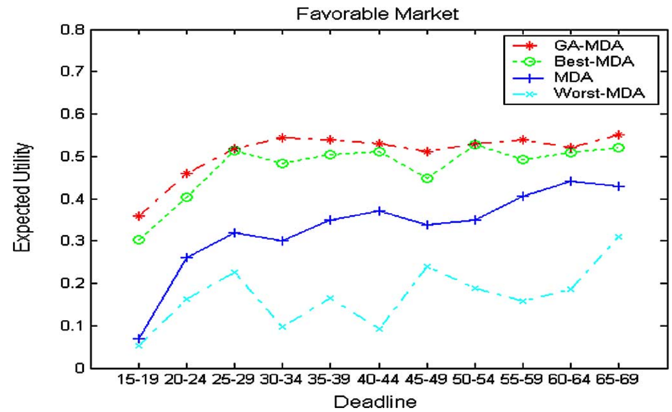

Fig. 2. Expected utility and deadline.

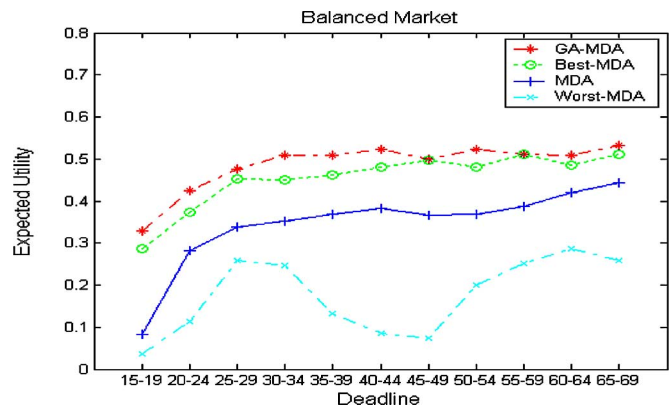

Fig. 3. Expected utility and deadline.

adopt the values of λ in Table III while the values of λ of MDAs were generated randomly. Even though experiments were carried out to compare the performance of GA-MDAs and MDAs in dense, moderate, and sparse markets, space limitations preclude all the results from being included in Section IV-F–H. In Section IV-F–H, only results comparing the performance of GA-MDAs with that of MDAs in dense markets are presented.

F. Observation 1

When both MDAs and GA-MDAs are subjected to different deadlines and market types, GA-MDAs achieved higher U_{expected} than MDAs.

It can be observed from Figs. 2–4 that for all deadlines and market types, GA-MDAs achieved higher utilities than MDAs (for example, in Fig. 2, when the deadline is between 35 and 39, the expected utilities are 0.54 for GA-MDAs, and 0.35 for MDAs, respectively). However, in unfavorable markets, the difference between the utilities of GA-MDAs and MDAs tapers. For example, in Figs. 2–4, when the deadline is between 35 and 39, the expected utilities are 0.54 for GA-MDAs and 0.35 for MDAs in favorable markets, respectively; 0.51 for GA-MDAs and 0.37 for MDAs in balanced markets, respectively; and 0.46 for GA-MDAs and 0.33 for MDAs in unfavorable markets, respectively. This is because in unfavorable markets, the bargaining positions of both GA-MDAs and MDAs are weaker and if

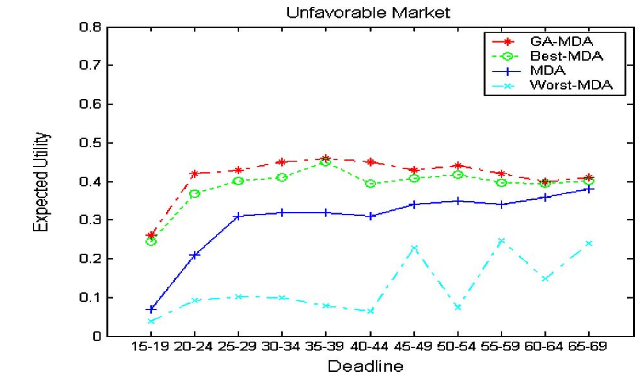

Fig. 4. Expected utility and deadline.

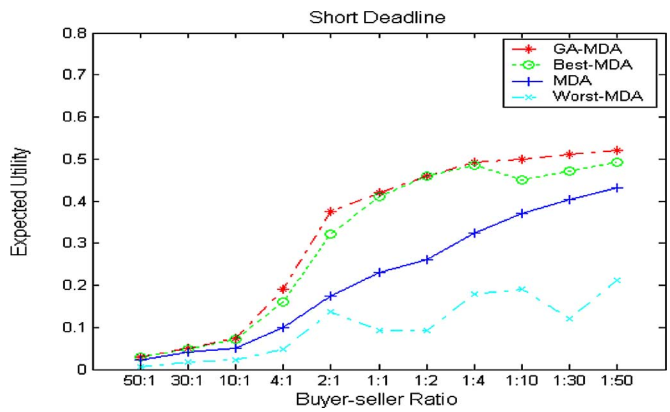

Fig. 5. Expected utility and buyer–seller ratio.

agreements are reached, both GA-MDAs and MDAs are likely to make relatively more concessions.

Additionally, it can also be observed from Figs. 5–7 that GA-MDAs achieved higher utilities than MDAs for all buyer–seller ratios and all deadlines. However, from Fig. 7, it can be observed that in favorable markets and when both types of agents have longer deadlines, the difference in their utilities tapers, particularly, for buyer-favorable markets such as buyer–seller ratios of 1:10, 1:30, and 1:50. Even though the strategies of both GA-MDAs and MDAs are designed for both buyers and sellers, for ease of exposition, the empirical results in this section are plotted from the perspectives of buyer GA-MDAs and buyer MDAs only. Hence, buyer–seller ratios such as 1:10, 1:30, and 1:50 are favorable for buyer GA-MDAs and buyer MDAs. In favorable markets and when both types of agents are subjected to longer deadlines, they have stronger bargaining positions and they are both likely to make less concessions.

From Figs. 2–7, it can be seen that GA-MDA generally achieved higher U_{expected} than Best-MDA, and the U_{expected} of Worst-MDA is much lower than that of MDA, Best-MDA, and GA-MDA.

G. Observation 2

When both MDAs and GA-MDAs are subjected to different deadlines and market types, GA-MDAs achieved higher R_{success} than MDAs.

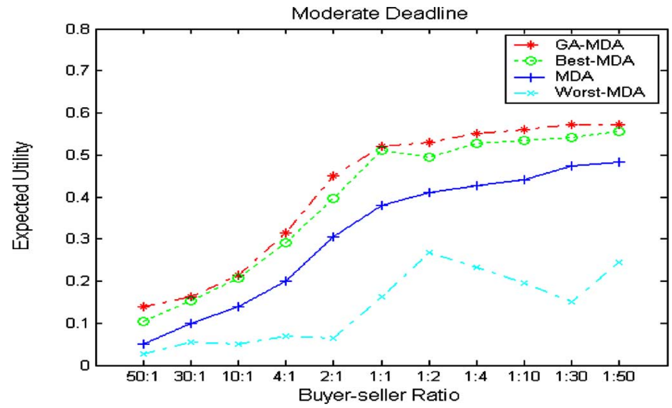

Fig. 6. Expected utility and buyer–seller ratio.

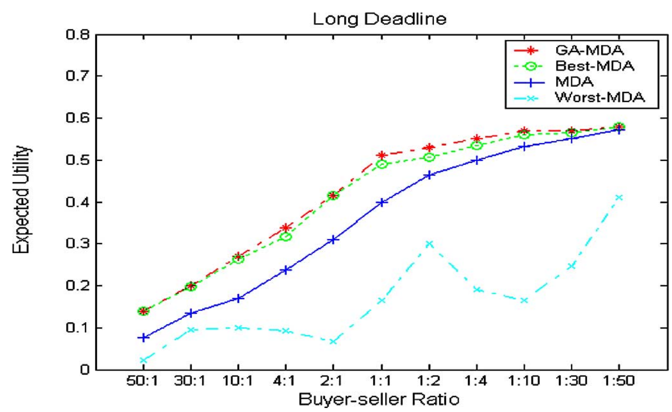

Fig. 7. Expected utility and buyer–seller ratio.

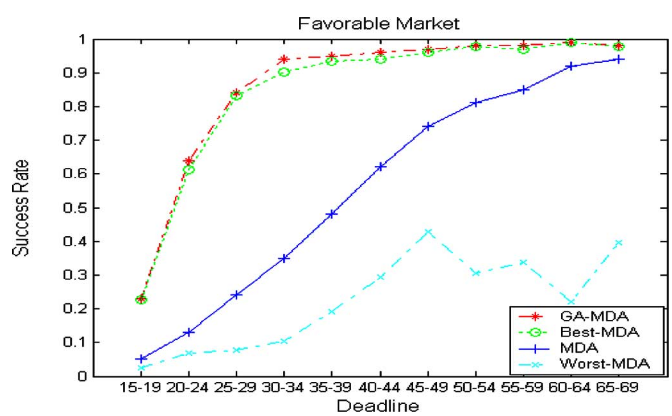

Fig. 8. Success rate and deadline.

It can be observed from Figs. 8–10 that for all deadlines and market types, GA-MDAs achieved higher R_{success} than MDAs. For example, in Fig. 8, the success rates are 0.95 for GA-MDAs and 0.48 for MDAs when the deadline is between 35 and 39. However, for very short deadlines (i.e., shorter than 20), R_{success} of both GA-MDAs and MDAs are relatively lower (Fig. 11). Under very extreme (or very adverse) trading conditions (e.g., with very short deadlines), it is extremely difficult for both GA-MDAs and MDAs to reach agreements. From Fig. 13, it can be observed that for longer deadlines (e.g., 50 and longer), R_{success} of both GA-MDAs and MDAs almost coincide

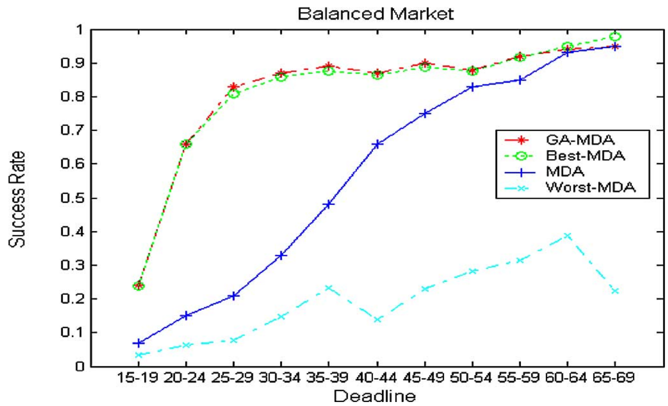

Fig. 9. Success rate and deadline.

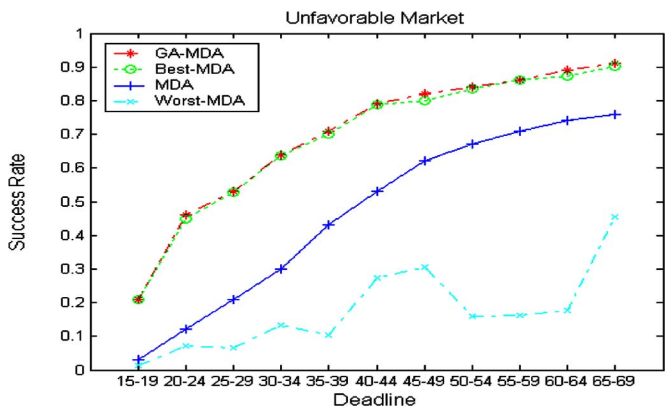

Fig. 10. Success rate and deadline.

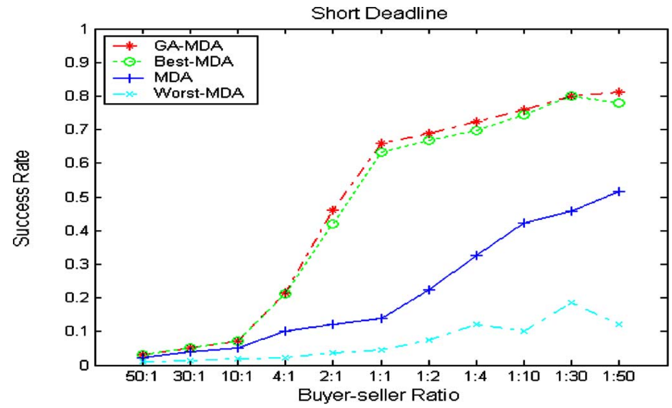

Fig. 11. Success rate and buyer–seller ratio.

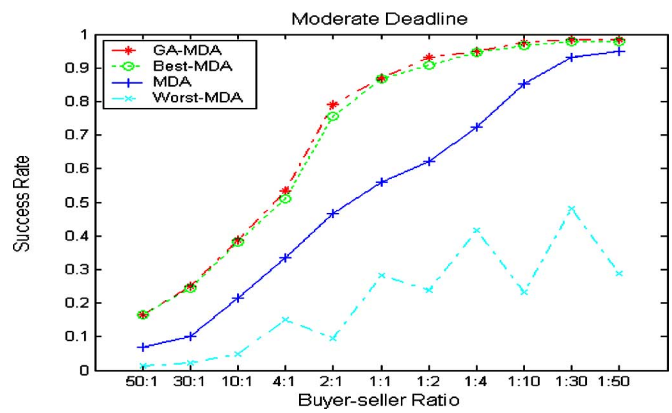

Fig. 12. Success rate and buyer–seller ratio.

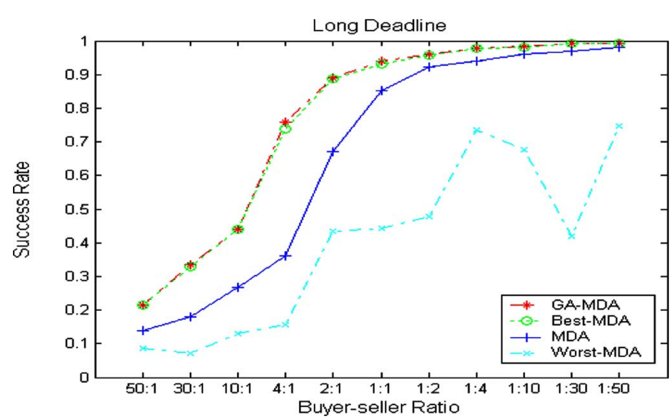

Fig. 13. Success rate and buyer–seller ratio.

(the same behavior is also recorded for deadlines longer than 70). With long deadlines, both GA-MDAs and MDAs have plenty of time for trading and they are both likely to complete deals successfully. Additionally, from Figs. 8–10, it can also be observed that in unfavorable markets, the difference in success rates between GA-MDAs and MDAs tapers. In unfavorable markets, both GA-MDAs and MDAs have weaker bargaining positions and it would be relatively more difficult for both GA-MDAs and MDAs to complete their deals successfully.

From Figs. 8–13, it can be seen that even though the success rate of Best-MDA is close to that of GA-MDA, the success rate of GA-MDA is still slightly higher than that of Best-MDA in most cases. However, the success rate of Worst-MDA is much lower than that of MDA, Best-MDA, and GA-MDA.

H. Observation 3

When both MDAs and GA-MDAs are subjected to different deadlines and market types, GA-MDAs generally take fewer negotiation rounds for reaching agreements than MDAs.

From Figs. 14–19, it can be observed that GA-MDAs generally achieved lower R_{time} than MDAs. For example, in Fig. 14, when the deadline is between 35 and 39, the average number of negotiation rounds are 23 for GA-MDAs, and 33 for MDAs, respectively. However, for very short deadlines (i.e., less than 40), the R_{time} of GA-MDAs is not significantly lower than the R_{time} of MDAs, especially when the deadline is less than 30 (see Figs. 14–16). With (very) short deadlines, both GA-MDAs and MDAs have very little time for trading and GA-MDAs did not outperform MDAs in terms of R_{time} for all buyer–seller ratios (see Fig. 17). With longer deadlines, GA-MDAs clearly outperformed MDAs in terms of R_{time} for all buyer–seller ratios (see Figs. 18 and 19).

From Figs. 14–19, it can be observed that GA-MDA generally took shorter average negotiation time than Best-MDA, and the

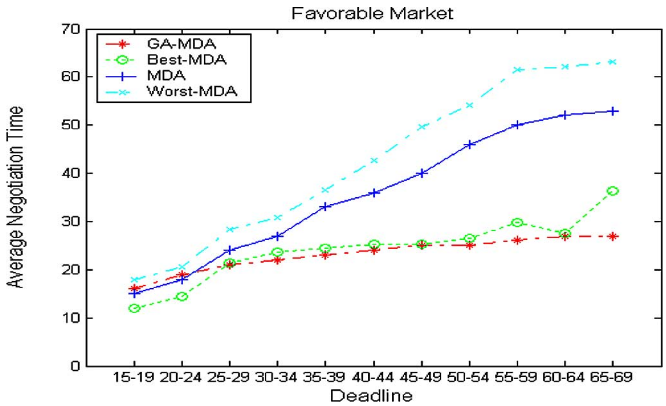

Fig. 14. Average negotiation speed and deadline.

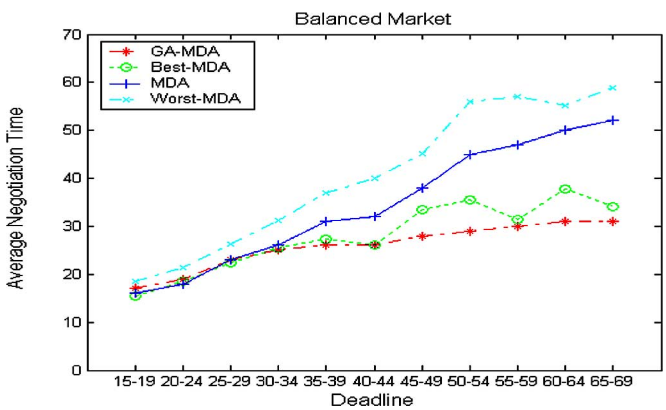

Fig. 15. Average negotiation speed and deadline.

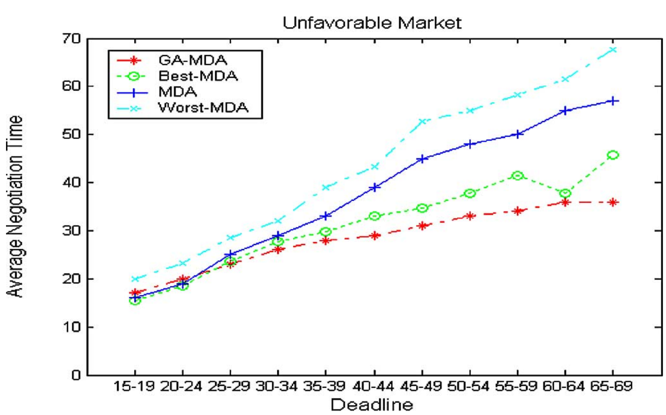

Fig. 16. Average negotiation speed and deadline.

average negotiation time of Worst-MDA is much longer than that of MDA, Best-MDA, and GA-MDA.

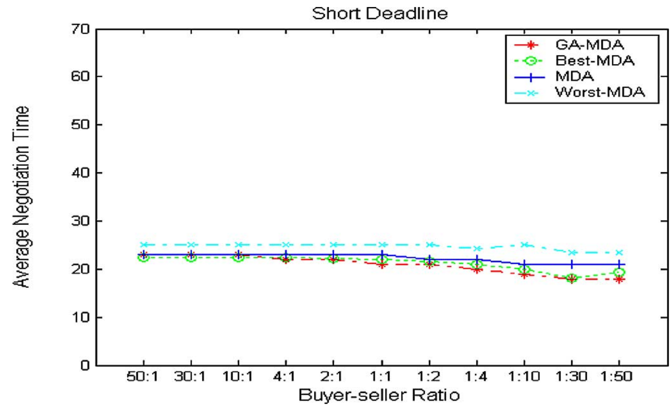

Fig. 17. Average negotiation speed and buyer–seller ratio.

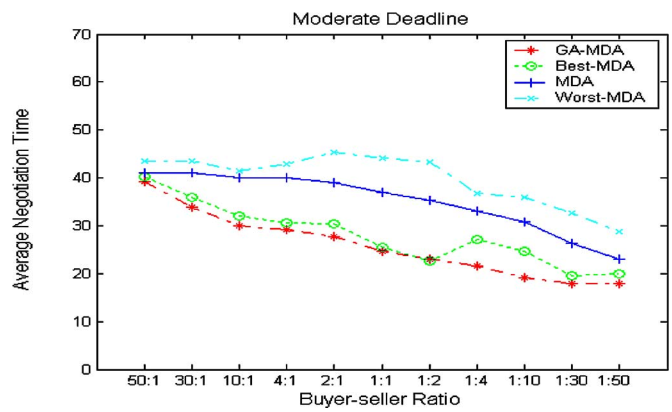

Fig. 18. Average negotiation speed and buyer–seller ratio.

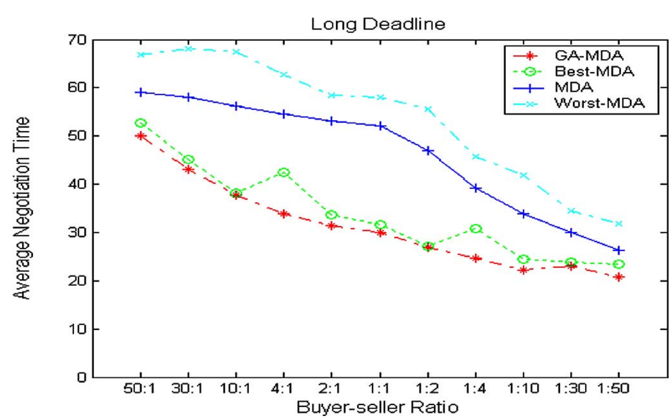

Fig. 19. Average negotiation speed and buyer–seller ratio.

V. Related Work

The literature of automated negotiation and negotiation agents (e.g., [10], [20], [33], [36], [37], [41]–[43]) forms a very large collection and space limitations preclude introducing all of them here. For a survey on negotiation agents for e-commerce and grid resource management, see [15], [24], and [40], respectively. This section only introduces and discusses related work on applying GAs to automated negotiation. In general, GAs are used to enhance automated negotiation in two ways: 1) GAs were used as a decision making component at every round, e.g., [22] and [23] and 2) GAs were used to learn the best strategies, e.g., [25]. The GA used in this work falls into the second kind.

Determining successful negotiation strategies: Matos *et al.* [25] have utilized GA for learning the most successful bargaining strategies in different circumstances (e.g., when an agent is facing different opponents). The negotiation model in [25] is based on the negotiation model in [10], in which an agent has a range of strategies based on time-dependent, resource-dependent, and behavior-dependent *NDFs*. Time-dependent *NDFs* consist of: 1) the *Boulware* tactic; 2) the *Conceder* tactic; and 3) the *Linear* tactic. Whereas the Boulware tactic maintains an agent's bid/offer until almost toward the end of trading (whereupon it concedes up to the reservation value), in the Conceder tactic, an agent rapidly concedes to its reservation value. Resource-dependent *NDFs* (consisting of impatient, steady, and patient tactics) generate proposals based on how a particular resource (e.g., remaining bandwidth) is being consumed. Agents become more conciliatory as the quantity of resource diminishes. In behavior-dependent *NDFs*, an agent generates its proposal by replicating (a portion of) the previous attitude of its opponent. An agent's strategy is based on a combination of the time-dependent, resource-dependent, and behavior-dependent *NDFs*. By placing different weightings for the time-dependent, resource-dependent, and behavior-dependent *NDFs*, different strategies can be composed. In [25], each agent is represented as a chromosome (or a string of fixed length). The bits of a string (or a chromosome) encode the parameters of an agent's strategy. In [25], agents negotiate over two issues, hence each of the strings encodes the two negotiation issues, the negotiation deadline, and the weighting of each of the combinations of tactics. In their GA, tournament selection is used to create the mating pool of the chromosomes that form the basis for the next population. The three basic genetic operators: reproduction, crossover, and mutation were used for generating new (and better) strategies. Whereas empirical studies seem to indicate the agents in [25] are generally effective in learning the best-response strategies for different circumstances, they are only designed to model bilateral negotiations, and notions of competition and opportunity are not explicitly modeled.

Evolutionary Learning and Adaptive Negotiation Agents: Lau *et al.* [22], [23] present an evolutionary learning approach for designing negotiation agents. Adopting GA as a heuristic search for deriving potential negotiation solutions, their negotiation agents adapt to changing behaviors of their opponents by learning about their preferences through their previous counter-offers. In [22], [23], negotiation agents are designed with the intuition of not only optimizing an agent's individual payoff but also striving to ensure that a consensus is reached. The authors represent a subset of feasible offers in a negotiation round as a population of chromosomes, and their GA approach evaluates the effectiveness of a negotiation solution using a fitness function that determines both: 1) the similarity of a negotiation solution to that of an opponent's proposal according to a weighted Euclidean distance function and 2) the optimality of the negotiation solution. In their GA approach, each chromosome encodes an offer using a fixed number of fields. The genes (fields) of each chromosome encode the unique identity of the chromosome, its fitness, and the attributes of an offer such as price, quantity, and shipment. Similar to [25], the GA in [22] and [23] utilizes the three genetic operators of reproduction, crossover, and mutation, and, at each iteration, either the tournament or Roulette-wheel selection is used to select chromosomes from the current population for creating a mating pool. Empirical results obtained by the authors seem to indicate that their GA-based negotiation agents can acquire effective negotiation tactics. Like agents in [25], the agents in [22] and [23] do not explicitly model outside options and market rivalry.

Tracking shifting tactics and changing behaviors: Krovi *et al.* [21] have devised a GA-based model for negotiation. Their work addressed the issue of tracking the shifting negotiation tactics and changing preferences of negotiators. In Krovi *et al.*'s [21] adaptive negotiation agents (ANAs), decision making of a negotiator is modeled with computational paradigms based on GAs. Novel features of *ANA* include: 1) the adoption of different tactics in response to opponents' tactics; 2) modeling the knowledge of opponents' preferences; 3) considering the cost of delaying settlements; 4) achieving different levels of goals in negotiation; and 5) considering the different magnitude of initial offers. The GA-based negotiation mechanism is used to model the dynamic concession matching behavior arising in bilateral negotiation situations. In [21], the set of feasible offers of an agent is represented as a population of chromosomes, and the "goodness" of each chromosome (i.e., each feasible offer) is measured by a fitness function derived from *Social Judgment Theory* (*SJT*). Similar to [22], [23], and [25], the three basic genetic operators of reproduction, crossover, and mutation were used in the GA in [21], and using a predefined number of iterations, the fittest chromosomes from the current population is selected as a tentative solution that represents the counter-offer. One of the disadvantages of *ANA* is that since the fitness function is based on *SJT*, an agent's evaluation of its opponent's counter-offer(s) may be subjective. Furthermore, agents in [21], like those in [22], [23], and [25], are only designed for bilateral negotiations.

Trade GA: Montano–Rubenstein and Malaga proposed an approach that employs a GA for finding solutions for multilateral negotiation problems involving multiple attributes [31]. In addition to the basic genetic operators such as crossover, and mutation, *Trade* GA is characterized by having a new genetic operator called *trade* for addressing problem specific characteristics. The trade operator models a concession making mechanism that is often used in negotiation systems. It simulates the exchange of a resource of one negotiator with a resource of another negotiator. When the trade operator is applied, participants (negotiators) and resources are randomly selected based on their willingness to trade. In [31], the performance of Trade GA is compared with traditional GA, random search, hill climbing algorithm, and nonlinear programming, and empirical results seem to suggest that Trade GA outperformed all the other approaches. However, like [21]–[23] and [25], Trade GA does not explicitly model market factors such as market rivalry and outside options.

Learning Negotiation Rules: In [26], GA is used to learn effective rules for bolstering a bilateral negotiation process. Unlike the work in [25] and [21]–[23] in which chromosomes are used to encode strategies and offers respectively, chromosomes in [26] represent (classification) rules. In [26], the fitness of a rule (choromosome) is determined by the frequency that the rule is used to contribute to a successful negotiation process (i.e., the number of times the rule is used to contribute to reaching a consensus). The basic genetic operators of reproduction, crossover, and mutation were used. Empirical results seem to indicate that genetically learned rules are effective in supporting users in several bilateral negotiation situations. The results in [26] also show that in a bargaining process, an effective negotiation rule is one that prescribes small step concessions and introduces new issues into the negotiation process. However, it is noted that in [26] only simple examples of bilateral bargaining (e.g, a two-party bargaining over house purchase) are used. Nevertheless, it is acknowledged that [26] is one of the earliest works that utilize a GA for bolstering negotiation support systems.

Bargaining With Discount Factors and Outside Options: Jin and Tsang [16] constructed a two-population coevolutionary system implemented using genetic programming (GP) for solving a two-player bargaining game. Each population modeled a strategy pool consisting of candidate solutions for each player to maximize its payoff from bargaining agreements. The strategies of the two players coevolved at the same time so that strategies that performed well were more likely to be chosen to undergo progressive modifications to be more competitive in forthcoming bargaining. Using a population size of 100, at each iteration, the GP in [16] utilized tournament selection for selecting chromosomes from the current population for creating a strategy pool, and crossover and mutation were used to operate on each strategy pool. An *incentive method* was used to handle constraints in [17] to define the fitness function for the individuals in the population. Empirical results in [16] showed that the GP generates solutions that can closely approximate game-theoretic solutions. Whereas a player in [16] can only either accept or reject an offer, a player in [17] chooses to end the bargaining by opting out (i.e., having the alternative to secede after rejecting an offer and take up an outside option). Jin and Tsang [17] modified the coevolutionary system in [16] to satisfy the additional requirement introduced by having outside options. In the co-evolutionary system in [17], outside option is an additional determinant of the bargaining outcomes. Even though the introduction of an additional determinant has made the bargaining problem more complex, empirical results in [17] show that the coevolutionary system in [17] can generate reasonably good strategies that approximate game-theoretic solutions. Whereas [16], [17], and this work adopt the same motivation that an evolutionary approach (e.g., GA or GP) may be an alternative to game-theoretic solutions for bargaining problems, a notable difference between this work and [16], [17] is that market rivalry was not explicitly modeled in [16] and [17].

VI. Conclusion

The novel feature of this work is designing and developing bargaining (negotiation) agents that can *both*: 1) evolve their optimal strategies and 2) adjust the amounts of concessions by reacting to changing market situations. Whereas several existing works (discussed in Section V) adopt GAs for evolving the most successful or best-response strategies for different negotiation situations, agents in these works mainly focused on bilateral negotiations and did not explicitly model the influence of market dynamics. Agents in this work are designed to evolve the best-response strategies for a wide variety of market situations (e.g., when agents are facing very stiff competitions, and when there is a very large number of agents in an e-market).

The contributions of this work are detailed as follows.

1) Although this work does not focus on inventing a new GA, it applies AFGA (Section III) to solve a difficult multiobjective optimization problem in multilateral negotiations (i.e., finding the best-response strategy that would generally optimize utility, speed, and success rate). This work is among the *few* works that applied GA for finding potential solutions in automated negotiation. Whereas [21]–[23], [25], and [26] used a GA for deriving potential solutions for two-party (bilateral) negotiations, this work adopts AFGA for evolving the best-response strategies in multilateral (many-to-many) negotiations. In addition, although both [25] and this work adopt time-dependent negotiation strategies, Matos' GA [25] evolves negotiation strategies that only optimize utilities. The AFGA in this work evolves negotiation strategies that optimize expected utilities, success rates, and negotiation speed, in response to different market situations. Furthermore, in the fitness function (Section III-B) of the AFGA in this work, both the constraints and preferences of GA-MDAs on the three objectives to be optimized (i.e., expected utilities, success rates, and negotiation speed) are modeled using a priority operation [9, p. 50], and by placing weights on the constraints. This allows different preferences and constraints to be programmed into GA-MDAs for different requirements in different applications. Finally, even though [31] applies GA for deriving solutions for multilateral negotiations involving multiple issues, in their setting, agents are cooperative agents. In this work, AFGA is applied to multilateral negotiations in which agents are competitive, and market factors such as market rivalry and outside options are explicitly modeled.
2) A testbed (Section IV-A) for evaluating the performance of GA-MDAs in different market situations was developed. The differences between the author's previous testbeds in [36], [38], [41], [42] and this work are as follows. The testbed in [41] consists of only MDAs, and it examined the impact of factors such as, deadline, buyer–seller ratio, and the difference in proposals between negotiation agents, on the negotiation outcomes of MDAs. The testbeds in [36] and [42] consist of both MDAs and EMDAs (enhanced MDAs with fuzzy decision controllers for relaxing bargaining terms), and they compared the performance of MDAs and EMDAs in terms of the tradeoff between success rates and expected utilities in negotiation. The testbed in [38] compares the performance of MDAs and EMDAs (in terms of success rates and utilities) with NDF [10] and *Kasbah* [4] agents in a grid-commerce environment. The testbed in this work compares the performance of GA-MDAs with MDAs (in terms of negotiation speed, success rates, and utilities) in a wide variety of market situations.
3) Empirical results (Section IV-F–H) show that in a wide variety of market situations (i.e., different market densities, market types, and deadlines), GA-MDAs generally achieved higher expected utilities and higher success rates,

and reached agreements using fewer negotiation rounds than MDAs. These results complement and supplement previous theoretical results obtained in [33], [34], and [37]. Even though Sim [34], [37] has proven that MDAs negotiate optimally by making minimally sufficient concession with respect to opportunity and competition (see [28, Lemmas 4.1 and 4.2, pp. 718–719]), there is no universal best-response time-dependent strategy (i.e., an appropriate value of λ) for different market situations. Given both the constraints and preferences of *GA-MDA*, different values of λ (i.e., different best-response strategies) of GA-MDAs can be evolved using the AFGA in this work (see Table III). To this end, this work elaborates and expands [34, Propositions 5 and 6, pp. 627–628]: "For larger τ (respectively, smaller τ), an MDA should adopt a larger λ (respectively, smaller λ)." For example, in Table III, it can be seen that in an unfavorable and moderately dense market, the best-response strategies for GA-MDAs are: 1) adopting a λ-value of 0.63 for a short deadline and 2) adopting a λ-value of 1.42 for a long deadline.

Whereas the AFGA in its present form is only used to evolve the best-response strategies of GA-MDAs to optimize utility, success rate, and average negotiation speed, perhaps, it may also be used to optimize other objectives in negotiation (e.g., optimizing resource utilization). Another possible future direction for this work is perhaps designing GAs that can perhaps converge faster by pruning the initial search space of GA-MDAs in some situations. For instance, for evolving the best-response strategies when GA-MDAs have longer deadlines, a search space of $[>1, 100]$ may be used, since in [34, Proposition 5, p. 627] prescribes the adoption of a conservative strategy (i.e., >1) when agents have plenty of time for negotiation. However, this may be done at the expense of obtaining agreements at a later stage and perhaps lowering the success rates of negotiation. Nevertheless, future experimentations are needed to verify these claims and this is among the list of agendas for future work.

In the problem setting in this work, real-world users adopt MDAs or GA-enhanced MDAs (GA-MDAs) to negotiate on their behalf, and each real-world user can potentially have his/her own different preferences and priorities for the utility, success rate, and negotiation speed. Hence, each real-world user can specify different weightings for each of the performance measures (i.e., utility, success rate, and negotiation speed). Using an AFGA enables agent designers to specify different preferences and priorities for different objectives. Whereas in their present form, GA-MDAs adopt an AFGA for evolving their negotiation strategies, one possible future work is to adopt multiobjective GAs such as those in [6]–[8] in place of the present AFGA in the current work. Whereas this work takes the first step in showing that GA-MDAs that adopt GA to evolve best-response strategies under different deadline constraints and market situations outperform MDAs that do not have the capability to adapt and evolve their strategies, another future enhancement of this work is to adopt a real-coded GA [13], [14] for evolving the strategies of GA-MDAs.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for their comments and suggestions for improving this paper.

REFERENCES

[1] C. W. Ahn and R. S. Ramakrishna, "Elitism-based compact genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 367–385, Aug. 2003.

[2] T. Back and M. Schutz, "Intelligent mutation rate control in canonical genetic algorithms," in *Proc. 9th Int. Symp. Found. Intell. Syst*. Piscataway, NJ, 1996, pp. 158–167.

[3] A. Chavez, D. Dreilinger, R. Guttman, and P. Maes, "A real-life experiment in creating an agent marketplace," in *Proc. 2nd Int. Conf. Practical Appl. Intell. Agents MultiAgent Technol.*, 1997, pp. 159–178.

[4] A. Chavez and P. Maes, "Kabash: An agent marketplace for buying and selling goods," in *Proc. 1st Int. Conf. Practical Appl. Intell. Agents Multi-Agent Technol.*, 1996, pp. 159–178.

[5] C. A. C. Coello, "An updated survey of ga-based multiobjective optimization techniques," *ACM Comput. Surveys*, vol. 32, no. 2, pp. 109–143, 2000.

[6] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer, 2002.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[8] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[9] D. Dubois and H. Prade, "Qualitative possibility theory and its applications to constraint satisfaction and decision under uncertainty," *Int. J. Intell. Syst.*, vol. 14, no. 1, pp. 45–61, 1999.

[10] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Int. J. Robot. Auton. Syst.*, vol. 24, no. 3–4, pp. 159–182, 1998.

[11] S. Fatima, M. Wooldridge, and N. R. Jennings, "A comparative study of game theoretic and evolutionary models for software agents," *Artif. Intell. Rev.*, vol. 23, no. 2, pp. 187–205, 2005.

[12] L. Fu and Y. Feng, "A fuzzy reasoning based bidding strategy for continuous double auctions," in *Proc. 2007 IEEE Int. Conf. Control Autom.*, pp. 1769–1774.

[13] F. Herrera, M. Lozano, and A. Sanchez, "A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study," *Int. J. Intell. Syst.*, vol. 18, no. 3, pp. 309–338, 2003.

[14] F. Herrera, M. Lozano, and J. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for the behavioral analysis," *Artif. Intell. Rev.*, vol. 12, no. 4, pp. 265–319, 1998.

[15] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge, "Automated negotiation: Prospects, methods and challenges," *Int. J. Group Decision Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.

[16] N. Jin and E. P. K. Tsang, "Co-evolutionary strategies for an alternating offer bargaining problem," in *Proc. IEEE Symp. Comput. Intell. Games*, 2005, pp. 211–217.

[17] N. Jin and E. P. K. Tsang, "Co-adaptive strategies for sequential bargaining problems with discount factors and outside options," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 2149–2156.

[18] G. Kennedy, *Field Guide to Negotiation*. Cambridge, MA: MIT Press, 1994.

[19] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[20] S. Kraus, "Automated negotiation and decision making in multi-agent environments," *Multi-Agents Syst. Appl.*, vol. 2086, no. 2, pp. 150–172, 2001.

[21] R. Krovi, A. C. Graesser, and W. E. Pracht, "Agent behaviors in virtual negotiation environments," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 1, pp. 15–25, Feb. 1999.

[22] R. Lau, "Towards genetically optimised multi-agent multi-issue negotiations," in *Proc. 38th Hawaii Int. Conf. Syst. Sci. (HICSS'05)*. Piscataway, NJ, Jan., p. 35c.

[23] R. Lau, M. Tang, O. Wong, S. Milliner, and Y.-P. Chen, "An evolutionary learning approach for adaptive negotiation agents," *Int. J. Intell. Syst.*, vol. 21, no. 1, pp. 41–72, 2006.
[24] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings, "A classification scheme for negotiation in electronic commerce," *Int. J. Group Decision Negotiation*, vol. 12, no. 1, pp. 31–56, Jul./Aug. 2003.
[25] N. Matos, C. Sierra, and N. R. Jennings, "Determining successful negotiation strategies: An evolutionary approach," in *Proc. 3rd Int. Conf. Multi-Agent Syst. (ICMAS-98)*, Jan., pp. 182–189.
[26] S. Matwin, T. Szapiro, and K. Haigh, "Genetic algorithms approach to a negotiation support system," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 102–114, Jan./Feb. 1991.
[27] H. Muhlenbein, "How genetic algorithms really work: mutation and hill-climbing," in *Proc. 2nd Parallel Problem Solving Nature*. Piscataway, NJ, 1992, pp. 15–25.
[28] M. J. Osborne and A. Rubinstein, *Bargaining and Markets*. New York: Academic, 1990.
[29] J. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. Cambridge, MA: MIT Press, 1994.
[30] C. D. Rosin and R. Belew, "New methods for competitive coevolution," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, 1997.
[31] B. Rubenstein-Montano and R. A. Malaga, "A weighted sum genetic algorithm to support multiple-party multi-objective negotiations," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 366–377, Aug. 2002.
[32] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.
[33] K. M. Sim, "A market-driven model for designing negotiation agents," *Comput. Intell., Spec. Issue Agent Technol. E-Commerce*, vol. 18, no. 4, pp. 618–637, 2002.
[34] K. M. Sim, "Equilibrium analysis of market-driven agents," *ACM SIGECOM: E-Commerce Exchange*, vol. 4, no. 2, pp. 32–40, 2003.
[35] K. M. Sim, "From market-driven agents to market-oriented grids," *ACM SIGECOM: E-Commerce Exchange*, vol. 5, no. 2, pp. 45–53, Nov. 2004.
[36] K. M. Sim, "Negotiation agents that make prudent compromises and are slightly flexible in reaching consensus," *Comput. Intell., Spec. Issue Bus. Agents Semantic Web*, vol. 20, no. 4, pp. 643–662, 2004.
[37] K. M. Sim, "Equilibria, prudent compromises, and the "waiting" game," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 3, pp. 712–724, Jun. 2005.
[38] K. M. Sim, "G-commerce, market-driven g-negotiation agents and grid resource management," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1381–1394, Dec. 2006.
[39] K. M. Sim, "Guest editorial: Special issue on agent-based grid computing," *Appl. Intell.*, vol. 25, no. 2, pp. 127–129, Oct. 2006.
[40] K. M. Sim, "A survey of bargaining models for grid resource allocation," *ACM SIGECOM: E-Commerce Exchange*, vol. 5, no. 5, pp. 22–32, Jan. 2006.
[41] K. M. Sim and C. Y. Choi, "Agents that react to changing market situations," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 33, no. 2, pp. 188–201, Apr. 2003.
[42] K. M. Sim and S. Wang, "Flexible negotiation agent with relaxed decision rules," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 3, pp. 1602–1608, Jun. 2004.
[43] K. M. Sim and E. Wong, "Towards market-driven agents for electronic auction," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 6, pp. 474–484, Nov. 2001.

Kwang Mong Sim (SM'07) received the B.Sc. (Hons.) degree (*summa cum laude*) from the University of Ottawa, Ottawa, ON, Canada, and the M.Sc. and Ph.D. degrees from the University of Calgary, Calgary, AB, Canada.

He is the Director of the Multiagent Systems Laboratory, Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, South Korea.

Dr. Sim is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C: APPLICATIONS AND REVIEWS and a member of the editorial boards of several other international journals. He was a Referee for several national research grant councils including the National Science Foundation and as a Keynote Speaker, the Honorary Chair, the Program Vice-Chair, and a Panel Speaker in many conferences. He is also a Guest Editor for five special journal issues on automated negotiation and grid computing, including the special issue on Game Theoretic Analysis and Stochastic Simulation of Negotiation Agents in the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C: APPLICATIONS AND REVIEWS. He is also a member of the IEEE Systems, Man, and Cybernetics' Society Technical Committee on Distributed Intelligent Systems.

Bo An received the B.Sc. and M.Sc. degrees in computer science from Chongqing University, Chongqing, China, in 2003 and 2006, respectively. He is currently working toward the graduation degree from the Computer Science Department, University of Massachusetts, Amherst.

His current research interests include autonomous agent and multiagent systems, especially in automated negotiation, resource allocation, and electronic commerce.