Toward Market-Driven Agents for Electronic Auction

Kwang Mong Sim and Eric Wong

Abstract—While there are several existing agent-based systems addressing the crucial and difficult issues of automated negotiation and auction, this research has designed and engineered a society of trading agents with two distinguishing features: 1) a market-driven negotiation strategy and 2) a deal optimizing auction protocol. Unlike some of the existing systems where users manually select predefined trading strategies, in the market-driven approach, trading agents automatically select the appropriate strategies by examining the changing market situations. Results from a series of experiments suggest that the market-driven approach generally achieved more favorable outcomes as compared to the fixed strategy approach. Furthermore, it provides a more intuitive simulation of trading because trading agents are able to respond to different market situations with appropriate strategies. By augmenting the auction protocol with a deal optimization stage, trading agents can be programmed to optimize transaction deals by delaying the finalization of deals in search of better deals. Experimental results showed that by having a deal optimization stage, the auction protocol produced generally optimistic outcomes.

Index Terms—Agent-mediated e-commerce, auction agent, automated negotiation.

I. INTRODUCTION

One of the most crucial issues in both conventional and electronic trading is for sellers and buyers to reach a consensus on pricing and other terms of transactions. The task of negotiation can often be difficult, time consuming and even frustrating for the average consumer [1]. In recent years, there have been several research initiatives in engineering agent-mediated electronic trading systems [2]–[9] that bolster negotiation (and other activities in electronic commerce such as information brokering, cataloging, advertising and payment). The main contribution of this research is designing and engineering a society of trading agents that help their human users negotiate terms of business transactions. In engineering negotiation and auction agents, the issues that are of interest include

1) the specification of the “rules of the game” (Rosenschein and Zlotkin [10]) or the agent interaction protocol;
2) the characteristics of agents;
3) the determination of the initial price range;
4) the management of the negotiation strategies.

Objectives: In empowering human users to make optimal purchase decisions, this research has designed and engineered an agent-based auction testbed by employing a market-driven approach [11]. In particular, the agenda of this research is to do the following.

1) Devise an auction protocol for a society of trading (buyer and seller) agents.
2) Formulate a utility theory for trading agents to determine and evaluate (initial) bids and offers.
3) Devise an approach for determining optimal trading strategies.
4) Design and engineer an agent-based testbed for supporting auction activities in multiple electronic marketplaces.
5) Evaluate the implemented auction testbed through a series of experiments.

The overall architecture of the agent-based auction testbed [see Fig. 1(a)] is explicated in Section II. Section III describes the various stages of the auction protocol. The highlights of this research: evaluation of bids using a utility function, evaluation of market situation and determination of optimal trading strategies, as well as optimizing deals are presented in Sections III-A–D, respectively. Experimental results and evaluations of the testbed are illustrated in Section IV. In Section V, the testbed is compared with some existing agent-based auction systems. Section VI concludes this paper.

II. AGENT-BASED AUCTION TESTBED

The agent-based auction testbed consists of a server agent that controls the operations of electronic marketplaces (which are virtual markets where electronic trading take place). Each electronic marketplace consists of a control agent and a set of trading agents (buyer and seller agents).

Electronic Marketplace: In the agent-based auction testbed, there can be many electronic marketplaces trading different goods or services. The control agent in a marketplace administers the admission and discharge of trading agents. It determines the commencement and termination of a marketplace and synchronizes events in the marketplace.

Server Agent: The server agent [see Fig. 1(a)] creates marketplaces and trading agents and provides users with an interface for entering profiles/preferences. It also has a local database that stores user profiles and information about the marketplaces. When a new trading agent is created, the server agent has to decide on an appropriate marketplace for the new agent. It will search the local database for marketplace(s) trading similar goods where the new trading agent can join the trading. If there is no matching marketplace, the server agent creates a marketplace trading the goods or services specified by the new trading agent.

Trading Agent: A trading agent is either a buyer agent or a seller agent that acts on behalf of a human user to trade with other trading agents in a marketplace. During negotiation, each
trading agent autonomously determines optimal strategies when dealing with other agents with the hope of maximizing the user's (and its own) utility.

**Implementation:** The agent-based testbed is implemented in Java (JDK 1.2.x) using a multitiered architecture approach [Fig. 1(b)] organized into three layers:

1) user interface layer;
2) information processing layer;
3) information storage layer.

In this implementation, a web browser is used as a user-interface, while the server, control and trading agents operate at the information processing layer. The information storage layer is implemented using a Microsoft SQL database server.

### III. AUCTION PROTOCOL

Executions of the stages of the auction protocol given in Fig. 2 are explained as follows.

1) **Market Creation:** Before a market commences, the server agent creates the marketplace as well as all the trading agents. When a marketplace is created, a control agent (CA) for the marketplace is also created. The CA initiates trading in the market when there are sufficient numbers of trading agents $N_{TA}$ or after a period of time $T_{start}$ (both $N_{TA}$ and $T_{start}$ are predefined values). At the beginning of an auction cycle, every trading agent sets its initial offer to the most desirable attributes specified by a human user. For example, a buyer may set the price range from $5 to $7, while a seller may set a range from $8 to $6. In this case, the buyer and seller agents start with initial bid/offer of $5 and $8, respectively.

2) **Compilation of Bids/Offerers:** In this phase, both buyer agents accept offers and seller agents accept bids under the coordination of the CA.

3) **Bids/Offerers Evaluation and Agent Selection:** In this phase, both seller agents evaluate bids and buyer agents evaluate offers using a utility function. The utility function ranks the bids and offers of trading agents according to various attributes specified. It provides a measure to guide
the selection of the preferred agent to trade with (see Section III-A).

4) **Determination of Market Situation:** Trading agents that did not reach a consensus in stage 4 will enter this phase. In this phase, trading agents evaluate the market situation using the approach in Section III-B.

5) **Making Concession:** Taking market situation into consideration, agents determine an optimal strategy to revise their bids and offers. The determination of strategies is explicated in Section III-C.

6) **Deal Optimization:** When two trading agents are engaged with bids and offers that favorably coincide with each other, they still have the option to postpone the final deal in the hope of receiving a better deal within a reasonable period. Details of the optimization of deals are presented in Section III-D. In addition, experimental results in Section IV-C suggest that the application of this feature is advantageous in close competitions.

A. Bids and Offers Evaluation

Bids and offers evaluation are based on a utility function $U$. In each cycle of trading, agents receive a list of offers or bids $(O_1, O_2, \ldots, O_n)$ from their trading partners. Each offer $O_i$ consists of a list of domain specific attributes $(a_1, a_2, \ldots, a_n)$. The value of each attribute $a_j$ is evaluated by a function $\delta$ that measures how close is $a_j$ to the desired value of attribute $j$. A larger (respectively smaller) value of $\delta$ indicates that $a_j$ is closer to (respectively further from) the desired value of attribute $j$. Each agent ranks all the offers or bids received using $U$ to determine the trading party with the most desirable offer or bid (i.e., the agent who’s offer or bid gives the maximum utility). The utility of an offer $O_i$ is defined as

$$U(O_i) = \sum_{j=1}^{n} w_j \delta(a_j)$$

where $w_j$ is a user specified weight factor for attribute $a_j$ and $\delta(a_j)$ measures how close is $a_j$ and the desired value of attribute $j$. $\delta(a_j)$ is defined as follows. Let the desired and reserved values of attribute $j$ be $D_j$ and $R_j$, respectively.

1) If $R_j < D_j$, then

$$\delta(a_j) = \begin{cases} 1, & \text{if } a_j < R_j \\ \frac{D_j - a_j}{D_j - R_j}, & \text{if } R_j \leq a_j < D_j \\ 0, & \text{if } a_j > D_j. \end{cases}$$

2) If $R_j > D_j$, then

$$\delta(a_j) = \begin{cases} 0, & \text{if } a_j > R_j \\ \frac{a_j - R_j}{D_j - R_j}, & \text{if } R_j \leq a_j < D_j \\ 1, & \text{if } a_j < D_j. \end{cases}$$

where 1) and 2) model the cases when an agent tries to decrease the value of an attribute to increase its utility and increase the value of an attribute to increase its utility. For example, a seller will model its utility of price using 1) while a buyer will use 2).

When trading agents are created, users are requested to rank the importance of all the domain attributes in the market such as price, credit period and expected delivery day(s) by entering the weight of each of the attributes. To facilitate direct comparison of the importance of different attributes, each $\delta(a_j)$ is normalized as follows:

$$0 \leq \delta(a_j) \leq 1 \quad \forall j \quad \text{and} \quad \sum_{j=1}^{n} w_j = 1 \quad \therefore \quad 0 \leq U(O_i) = \sum_{j=1}^{n} w_j \delta(a_j) \leq 1.$$

**Example 1:** The following scenario illustrates how agents evaluate bids/offers based on $U$. Using only two domain attributes, but without lost of generality, Price ($P$) and Delivery Day ($D$) are considered in the market place. A user who can absorb higher price but requires rapid delivery of the goods may place more emphasis on $D$ and specifies the weight of $P$ and $D$ as 0.3 and 0.7, respectively. A buyer agent $A_B1$ representing the user evaluates the list of received offers based on the user specified weights. Suppose that two offers $O_A = \langle 5, 3 \rangle$ and $O_B = \langle 6, 2 \rangle$ are received (where the first and second attributes represent $P$ and $D$, respectively). $A_B1$ prefers $O_B$ over $O_A$ since $O_B$ requires only two delivery days while $O_A$ needs three days. However, the converse is true if there is another buyer $A_B2$ that is more concerned about price.

An algorithm to determine if all the attributes of trading terms coincide is given in Fig. 3. The issue of matching the terms of trading agents along multiple attributes is viewed as satisfying the constraints of these terms. Trading partners with more matching terms and attributes are evaluated more favorably. The algorithm for evaluating a list of offers (bids) and selecting the preferred seller (buyer) agent is given in Fig. 4.

B. Evaluate Market Situation

In determining the strategies for trading among agents, this research employs a market-driven approach [11]. The market situation at time $t$ is determined by

1) the number of seller agents $S_t$;
2) the number of buyer agents $B_t$;
3) the number of agents interested in the agent’s current offer $I_t$;
4) the closing time of the marketplace $T$.

Factors that favor an agent’s trading opportunity in an operational market include long remaining market time, decreasing competitions and increasing attractiveness of the agent’s offer. The notations of competition and attractiveness are explicated below. A market situation is quantified with a range of 0 to 1 where 1 indicates the worst condition and 0 indicates the best situation. Let $M(t)$ be the market situation for trading cycle $t$

$$M(t) = 1 - \frac{(I_t - t/T) + C(t) + A(t))}{3}$$

where

$$C(t) = \begin{cases} \frac{B_t}{S_t}, & \text{for seller agent} \\ \frac{S_t}{B_t}, & \text{for buyer agent} \end{cases}$$

$$A(t) = \begin{cases} \frac{I_t}{S_t}, & \text{for seller agent} \\ \frac{I_t}{B_t}, & \text{for buyer agent}. \end{cases}$$

The factor $T - t/T$ measures the relative remaining time of the marketplace. If an agent has been running for a long time...
Fig. 3. Algorithm for matching attributes of offers and bids.

(i.e., when $T - t/T$ is small), it implies that the agent has difficulty in engaging trading partners. $C(t)$ reflects market competition at time $t$ and is measured by the ratio of the number of buyer (seller) agents to the total number of agents if the agent is a seller (buyer) agent. Smaller values imply higher competition. For instance, if a buyer agent determines that $C(t) = 0.3$, it
implies that the agent expects vigorous competitions from other buyer agents. One of the most significant factors that favors an agent’s trading is the attractiveness of the agent’s offer \( A(t) \). \( A(t) \) is measured by the ratio of number of agents interested in the agent’s offer to the total number of potential trading partners. When an agent’s offer is losing attractiveness, that is, when no agent is interested in its offer (when \( I_t = 0 \)), the agent should react more vigorously and rapidly by giving a more compelling offer with the hope of luring potential trading partners. If the agent’s current offer has already generated some interests from other agents (when \( I_t \geq 1 \)), then the agent is in a good position to deal with its trading partners. The agent can slowly revise its own offer while waiting for its trading partners to compete with each other. The agent will only engage the trading partner that gives the best offer. In this case the agent is likely to obtain a more profitable deal.

Since smaller values of the three factors (remaining market time, competition factor and attractiveness indicate degrading market situation), \( M(t) \) is computed by taking the complement of the sum of the three factors. An experiment was carried out to demonstrate how trading agents react to changing market situations by employing different trading strategies. The results are reported in Section IV-A.

C. Determination of Strategies

The testbed described in this paper has a mechanism for determining the appropriate trading strategies in response to different market situations. As in most existing agent-based auction systems [3]–[5] each strategy corresponds to and can be implemented by different decay function. A decay function governs the rate at which an agent relaxes the value(s) of its next offer or bid. When a market commences, trading agents begin to place their initial bids and offers according to the most desirable attributes specified by their human users. As trading progresses, if an agent is still unable to attract and engage potential trading partners, it gradually relaxes the values in the bids or offers. The decaying rate of an attribute can often be described using some mathematical functions. In Kasbah [4], [5], for example, trading agents provide users with a range of strategies to select from and a selection from these strategies is essential a choice of different simple algebraic functions such as linear or quadratic functions. Each algebraic function formulaulates a decay magnitude to suit a user’s specification. By having users manually select the adjustment magnitude of an attribute, previous approaches do not take into consideration the dynamics of a changing environment in a marketplace. For instance, if a seller agent’s offering price has attracted a lot of attention, continuingly reducing the price in every trading cycle may not be prudent. This research adopts a reactive approach in automating the determination of optimal trading strategies according to the market situation. Trading agents revise their strategies (i.e., the magnitude of adjusting their offers/bids) based on the current market situation and eagerness (of their human users) to trade without user intervention. The adjustment magnitude in auction cycle \( t + 1 \), denoted by \( D(t+1) \), is computed from the market situation perceived at cycle \( t \)

\[
D(t+1) = \frac{M(t) + E}{2} \times D_{\text{max}}
\]

where

- \( M(t) \) market situation at time \( t \);
- \( D_{\text{max}} \) maximum adjustment scale;
- \( E \) user’s eagerness to trade.

The value of the eagerness factor \( E \) is supplied by the user. While \( E \) is an indication on how keen a human trader is in providing (or acquiring) a product (or a service), \( D_{\text{max}} \) is dependent on \( T \). The magnitudes of both the eagerness (\( E \)) and the maximum adjustment scale (\( D_{\text{max}} \)) are domain specific. Several experimental simulations were carried to measure the effect of \( E \) on the strategies of agents and the trading outcome (see Sections IV-B and D). In these experiments, the marketplace has a maximum duration of 1000 auction cycles (\( T = 1000 \)) and the value of \( E \) is arbitrary defined in the range of 0.1 to 1.0 while the value of \( D_{\text{max}} \) is set to 20. An algorithm for a market-driven agent to determine its strategy for making concession is given in Fig. 5.

D. Optimizing Deals

An algorithm for deal optimization is given in Fig. 6. If two potential trading agents are interested in each other’s offer, the trading process for both parties continues until they reach a consensus on all terms of transaction. However, the intuition is that sometimes it may be prudent to look for and consider other more promising opportunities that may arise after trading partners reached an agreement. To optimize deals, one approach that trading agents can adopt is not to finalize deals immediately but proceed to a deal optimization stage. The agent-based auction testbed supports this deal optimization stage where two agents can keep their offers unchanged but continue to receive offers from other agents. If there are no better deals after a reasonable period of time, the two agents will finalize the transaction.

Example 2: An example to demonstrate the feature of trading with deal optimization is shown in Figs. 7 and 8. Fig. 7 showed a trading situation without deal optimization. In this situation, there were two seller agents \( S\text{eller}_1 \) and \( S\text{eller}_2 \) and one buyer agent \( B\text{uyer}_0 \). \( S\text{eller}_1 \) has a starting offering price that is relatively higher than that of \( S\text{eller}_2 \), but unknown to \( B\text{uyer}_0 \). \( S\text{eller}_1 \)’s best price (lowest) price is lower (hence, more attractive) than \( S\text{eller}_2 \)’s. Initially, \( S\text{eller}_2 \)’s offer appeared to be comparably more attractive than \( S\text{eller}_1 \)’s. As trading commences, all three agents started to adjust their offers before a consensus is reached. Since \( S\text{eller}_2 \) set a lower price, it tends to approach \( B\text{uyer}_0 \)’s price at a more rapid rate than \( S\text{eller}_1 \). In the sixth cycle, \( B\text{uyer}_0 \) and \( S\text{eller}_2 \) seemed to reach a consensus as their price coincide. Without consideration of deal optimization, \( B\text{uyer}_0 \) will finalize the deal with \( S\text{eller}_2 \), even though \( S\text{eller}_1 \)’s offer would have been more attractive if \( B\text{uyer}_0 \) waited for two more cycles or longer (see the projection in dotted lines).

Under the same trading situation, the trading outcome of \( B\text{uyer}_0 \) would have been different if it had postponed its final deal. This is shown in Fig. 8. As in Fig. 7, \( S\text{eller}_1 \)’s offer is initially more attractive than \( S\text{eller}_2 \)’s. In the early stage of trading, the pattern of making concessions for all three trading agents were similar to Fig. 7, until the sixth cycle. The area in the big circle is when \( B\text{uyer}_0 \) and \( S\text{eller}_2 \) seemed to reach a consensus on price and this is when the optimizing mechanism
may take effect. This time, even though Buyer0 and Seller2 seemed to have reached similar terms, Buyer0 has delayed finalizing the deal in the hope that Seller1 may make a better offer reasonably quickly. During the waiting period, Seller1 kept revising its offer while the price of Buyer0 and Seller2 remained unchanged. Seller1 is gradually catching up with the offers of Seller2 and finally surpassed Seller2’s offer in the ninth cycle. Consequently, Buyer0 engaged with Seller1 at a price below $6.50 (Fig. 8) rather than with Seller2 at a price of $6.75 (Fig. 7). This example suggests how trading agents may benefit from deal optimization by postponing their final deals if better offers arrive quite rapidly.

### Function: Making Concession
**Input:** Buyer agent’s bid in the current cycle  
**Output:** Buyer agent’s current bid in the next cycle

/*This algorithm specifies the procedure for determining the current market situation and adjusting the buyer agent’s offer based on the market situation. It detailed stages S4 and S5 in Fig. 2. The corresponding algorithm from a seller’s perspective is identical. */

Let $t$ be the current trading cycle of the market time  
Let $B_A(t) = \{a_1, a_2, \ldots, a_n\}$ be the buyer agent’s bid at $t$  
And $B_A(t+1)$ be the buyer agent’s bid at $t + 1$  
Let $P$ be the preferred seller agent and $O_p(t) = \{b_1, b_2, \ldots, b_l\}$ be $P$’s offer at $t$  
Let $\varepsilon$ be an arbitrary small values of tolerance

**Begin**  
Set $B_A(t+1) = \emptyset$  
Compute market situation  
$$M(t) = 1 - \frac{C(t) + A(t)}{T - \frac{t}{T} + C(t) + A(t)}$$
where $t, T, C(t)$ and $A(t)$ are as defined in section 3.2

Compute adjustment factor $D(t+1) = \frac{M(t) + E}{2} \times D_{max}$  
where $E, D_{max}$ are as defined in section 3.3.

/*If attribute $i$ of the buyer agent’s bid is different from the preferred seller agent’s offer, the buyer agent adjusts its value of $a_i$.*/

If the adjustment function is an increasing function $D(t+1)$ is added to $a_i$, if it is a decreasing function, $D(t+1)$ is subtracted from $a_i$. */

For each $a_i \in B_A(t) = \{a_1, a_2, \ldots, a_n\}$  
If $|a_i - b_j| \geq \varepsilon$ Then  
Set $a_i = a_i \pm D(t+1)$  
End-if  
Set $B_A(t+1) = B_A(t+1) \cup a_i$  
End-for  
Return $B_A(t+1)$  
End

**Fig. 5. Algorithm for adjusting bids.**

### Function: Deal Optimization
**Input:** The preferred agent  
**Output:** Outcome of Deal Optimization

/*This algorithm specifies the procedure for deal optimization from a buyer agent’s perspective. It corresponds to stage S6 in Fig. 2. The corresponding algorithm for a seller agent is identical. */

Let $P$ be the preferred agent and $O_p$ be the offer of $P$.  
Let $\tau$ be current waiting cycle  
Let $T_p$ be the period of deal optimization  
Let $O_L$ be the list of attractive offers from seller agents  
Let $N_e$ be the number of agents that satisfied condition 1 in definition 1 of Section 3.4  
Let $\beta$ be an arbitrary value as defined in definition 1 in Section 3.4.

**Begin**  
If $N_e \geq \beta$ Then  
/*condition to enter deal optimization stage*/  
Set Predealing Status = true  
Set $\tau = 0$

Compute $T_p = \frac{1}{E} \times N_e$ where $E$ is as defined in section 3.3 and $N_e$ is as defined 1 in Section 3.4.

While $\tau < T_p$ and Predealing Status = true  
Continue to receive and evaluate offers  
/*When $P$’s offer is no longer attractive, it implies that another seller agent has made a better offer during the deal optimization stage. Hence, the buyer agent exits the deal optimization stage, returns to stage S3 and trading continues. */

If $P \notin O_L$  
Set Predealing Status = false  
End-If  
Set $\tau = \tau + 1$  
End-While  
Else  
Set Predealing Status = true  
End-If  
Return Predealing Status  
/*When the Predealing Status = true, it implies either:  
i. the deal optimization stage was bypassed because there was no close competition  
ii. the deal optimization stage was executed and there was no better deal*/

**End**

**Fig. 6. Algorithm for optimizing deals.**

The process for optimizing deal consists of two stages.  
1) **Decision to Enter the Deal Optimization Stage:** In the first stage, a trading agent decides whether to enter the deal optimization stage by determining if there is close competition among trading partners [i.e., trading partners which have offers (or bids) that rival the best offer (bid)]. For example, if a seller agent receives offers of $90, $92 and $94 from three buyer agents, then it appears that there are close competitions among the buyer’s trading partners as the difference of the three offers are sufficiently close to rival each other. In this example, the buyer would be more inclined to enter the deal optimization stage.  
2) **Deal Optimization Stage:** In the second stage, a trading agent determines if the current deal is better or worse than the previous deals. If a better deal is found, the trading agent will update its offer and re-enter the deal optimization stage. If no better deal is found, the trading agent will exit the deal optimization stage and continue with its next task.
Fig. 7. Trading without deal optimization.

stage. If the same seller agent receives offers such as $90, $20 and $25 then it is clear that the other two offers are unlikely to rival the best offer in the next (several) trading cycle(s). In this case, it does not seem prudent for the seller agent to postpone to completion of the deal. A more formal interpretation of close competition is given in Definition 1.

Definition 1: For a given set of trading agents \( \{a_1, a_2, \ldots, a_n\} \), close competition exists if

\[
\exists a_i \left[ U(O_{a_i}) - U(O_{\text{best}}) \right] \leq \alpha
\]

where

- \( O_{a_i} \) offer made by agent \( a_i \);
- \( O_{\text{best}} \) best offer;
- \( U(O) \) utility of an offer \( O \);
- \( \alpha \) depends on \( C(t) \).

A small value of \( C(t) \) indicates that an agent faces more competition (has many competitors—see Section III-B) and a smaller value of \( \alpha \) is preferred. Experiments that simulate different degrees of competition (with different \( C(t) \)) as shown in Fig. 9, showed that for situation with tight competitions (e.g., \( C(t) = 0.2, 0.3, 0.4 \)), an agent achieved higher utilities for smaller values of \( \alpha \) (e.g., \( \alpha = 0.01 \)). For larger values of \( C(t) \) (i.e., few competitors), agents performed better with larger values of \( \alpha \) (e.g., \( \alpha = 0.04 \) or \( 0.05 \) when \( C(t) = 0.7, 0.8, 0.9 \)). The above coincides with the intuition that it is less prudent to postpone a deal when there are many competitors unless there are other very attractive bids/offers that are close enough to rival the best bid/offer.

\[ N_a \geq 1 \], where \( N_a \) is the number of agents satisfying the above equation in Definition 1. Close competition exists when there is at least one agent with bid/offer that rivals the best bid/offer.

2) Determining the Period of Deal Optimization \( T_P \):

\[ T_P = \frac{1}{E} \times N_a \]

where \( N_a \) is the number of agents satisfying the first condition in 1), and \( E \) is the user’s eagerness (Section III-C). Since a larger value of \( E \) indicates that a user is more keen to complete a deal, the period of deal optimization will be shorter. Hence, \( T_P \) is determined by taking the product of \( N_a \) and the reciprocal of \( E \).

A series of experimental simulations (explicated in detail in Section IV-C) suggests that trading agents achieved higher utilities in close competitions if they enter the deal optimization stage.

IV. EVALUATION AND EXPERIMENTATION

To demonstrate the features of market-driven agents and to evaluate the effectiveness of the market-driven strategy, a series of experiments are carried out. The results are reported in Sections IV-A to D.

A. Reacting to Changing Market Situations

This experiment examines the effect of changing market situation \( M(t) \) on agents. Since all agents experience the same time constraint (i.e., \( T = t/T \)), it seems prudent to examine the effect of market situation by considering different buyer–seller ratios which affect both \( C(t) \) and \( A(t) \) (see Section III-B). For ease of demonstration (but without loss of generality), two domain-attributes price and delivery days were considered. Since market-driven strategies are designed for both buyers and sellers, it suffices to demonstrate the effect of market situation on seller agents without loss of generality. The experiment was conducted with four seller agents and the number of buyer agents varied from one to seven. Since this experiment
examined the effect of market competition (i.e., $C(t)$ and $A(t)$, which depends on buyer–seller ratio), all buyer agents are programmed with the same initial offers and eagerness $E$ to minimize their effect on the trading process (the effect of $E$ is discussed in Section IV-B). Experimental results (Fig. 10) showed that as the number of buyer agents increased from one to seven, seller agents’ average utility also increased. The results coincide with the intuition of trade competitions in real life; an agent achieves higher utility (hence, higher user’s satisfaction) when there are more trading options (more trading partners). Without considering $M(t)$, agents’ concession rate largely depends on predefined functions which may not be the most appropriate for a given market situation (see Section IV-D).

B. Effect of Eagerness

This experiment demonstrated how eagerness $E$ can generally influence an agent’s strategy in trading (the rate at which the agent relaxes its offers/bids) and the outcome of the trading (i.e., the agent’s utility). In this research, market-driven strategies are designed for both buyer and seller agents; thus, without loss of generality, it suffices to demonstrate the effect of eagerness on seller agents. Six sets of simulations were carried out, with one seller agent and a varying number of buyer agents from one to six. In each simulation, $E$ for all buyer agents were set to 0.1, 0.2, 0.4, 0.6, 0.8 and 0.95, respectively. The results are shown in Fig. 11. From Fig. 11, it can be observed that the seller agent achieved higher utility for large value of $E$ (e.g., $E = 0.95$, when buyers are more eager) than for smaller $E$ (e.g., $E = 0.1$, when buyers are less keen). For example, when there were six buyers with $E = 0.1$, the seller’s utility was approximately 0.6. For the same number of buyer agents but with $E = 0.95$, the seller’s utility was close to one. Furthermore, in another instance, even though when there were just three buyer agents with $E = 0.95$, the seller agent achieved a utility of about 0.7, which is higher than the previous instance where there were six buyers with $E = 0.1$. The results show that an agent trading with fewer number of trading partners, but with higher eagerness achieved higher utilities than with larger number but less eager trading partners. In summary, this experiment demonstrated that $E$ is one of the other factors (other than buyer–seller ratio) that affect its rate of concession and trading outcome. Without considering $E$, the seller agent was expected to achieve a higher utility as the numbers of buyer agents increased, producing increasingly favorable market situations (see Section IV-A).

C. Trading With Deal Optimization

An experiment to investigate the trading behaviors of agents using deal optimization was conducted. It demonstrated how trading agents adjusted the deal optimization period under different competitive situations. In this experiment, there was one buyer agent and the number of seller agents varied from one to five. To simulate close competitions, all seller agents have identical parameters. For instance, they made the same initial offers and have same eagerness ($E = 0.7$). Consequently, all the seller agents reacted very similarly and the offers they made were close to each other’s. An interpretation of close competition was given in the first equation of Definition 1. On detecting close competitions, the buyer agent extended the number of deal optimization cycles in the hope of optimizing its utility. The number of deal optimization cycles for different competitive situations were plotted in Fig. 12. The utilities of the buyer agent for the corresponding deal optimization cycles under the corresponding competitive situations were plotted in Fig. 13. To simulate a loose competition environment, the sellers’ eagerness factors were scattered between 0 and 1 and hence seller
agents adjusted their offers at different rates. In loose competitions, the buyer agent reduced the number of deal optimization cycles as shown in Fig. 12. The results in Fig. 13 suggest that a trading agent can achieve higher utility in close competition if it increases the number of deal optimization cycles (allow longer period for the deal optimization stage). In loose competition, a trading agent tends to achieve higher utility if lesser deal optimization cycles were used. This seems to correspond to the intuition that a trader is more likely to postpone the finalization of a deal (for a longer period) when there are fierce competitions from more trading partners.

D. Fixed Strategy versus Market-Driven Strategy

This experiment compared the performance of the fixed strategy and the market-driven strategy as well as the effect of eagerness. Three sets of simulations were carried out, each with different values (0.5, 0.7, and 0.9) of sellers’ eagerness. In each of the simulations, there were six buyer agents, each with similar initial offer and eagerness of 0.7. To simulate different market situations (generating different values of $1 - M(t)$), the number of seller agents was varied from one to ten. Each seller agent experienced more competition (has higher value of $C(t)$ as described in Section III-B) as the number of seller agents increased. The seller agents' average utility $U$ (user’s satisfaction) was determined for different market situations $M(t)$ and $U$ was plotted against $1 - M(t)$ as shown in Fig. 14. In Fig. 14, a larger value of $U$ indicated a higher user’s satisfaction and a larger value of $1 - M(t)$ is an indication that the seller has favorable market position. In the fixed strategy approach, trading agents reduced or increased their bids/offers by making constant adjustment based on predefined functions (such as linear, quadratic, or cubic functions) until a consensus is reached, without considering market conditions. Consequently, the fixed strategy approach, produced relatively constant $U$ even in the case when the agent is in a better market situation [when $1 - M(t)$ is larger]. Employing the market-driven approach, agents adjusted their offers/bids dynamically based on the evaluated market situation and hence produced more intuitive results. In general, using the market-driven approach, seller agents generated higher $U$ as compared to the fixed strategy approach. With buyers that are not so keen (e.g., with $E = 0.5$), seller agents achieved higher $U$ using market-driven strategies than with fixed strategies. However, in not so favorable market situations [when $1 - M(t)$ is small], an agent that is keen to complete a deal (i.e., having higher value of $E$), may achieve lower $U$. For example, with $E = 0.7$ and $E = 0.9$, although the utilities achieved using the market-driven strategy were generally better, it has comparatively lower utilities when $1 - M(t) = 0.2$ and $0.3$, respectively. While a seller agent is faced with more competition (as there were increasingly more buyer agents), it is also keen to complete the deal. Since the fixed strategy does not consider the notion of $E$, the combined effect of eagerness and market competition does not influence its performance. In summary, while the market-driven strategy does not always perform better than the fixed strategy, it is generally better and it provides a more intuitive modeling of trading behaviors in real life.

V. RELATED WORK

Some of the agent-based auction and negotiation systems that are related to this research include Fishmarket [2], AuctionBot [3], Kasbah [4], [5], Tete-a-Tete [6], [7], the case-based negotiation (CBN) framework [8], [9], and adaptive negotiation agent [12]. Fishmarket defined and implemented trading scenarios inspired by traditional fish market auction. Based on downward bidding or Dutch auction protocol, a market consists of seller and buyer agents, market intermediaries such as an auctioneer that calls prices in descending order and a market boss that declares the start and closing of auction. At every bidding round, an auctioneer quotes offers downward from the starting price fixed by the seller. For each quoted price, there can be

1) no bidder;
2) one bidder;
3) multiple bidders.

When there is no bidder, the auctioneer decrements the price if the reserved price has not been reached. If there is one bidder, the product is sold if the bidder has sufficient credit. When there are multiple bidders, the auctioneer resolves the tie by restarting a new round of bidding at a higher price. While seller and buyer agents may be encoded with different bidding strategies, trading is largely governed by the auction conventions that determines the methods in resolving ties and quoting the next offer.
AuctionBot is an Internet auction server that can be viewed as a generalization of the trading convention in Fishmarket. It supports a variety of auction types including: Dutch auction, English outcry auction, First Price Sealed Bid auction, Vickrey auction and Continuous Double auction. This flexibility is achieved by decomposing auction design space into a set of parameters. Some of these parameters include bidding restrictions (e.g., the number of buyers permitted), allocation policies (e.g., methods for resolving ties) and auction events (e.g., clearing time). New auctions are created for trading products by selecting the auction type and by specifying and adjusting these parameters. Bids among buyers and sellers are deliberated according to the multi-lateral distributive negotiation protocols of the created auction. AuctionBot manages and enforces agents’ bidding according to specified parameters. The allocation policy largely determines which agents transact and at what price(s), as a function of the bids. Although AuctionBot’s application interface supports the creation of personalized trading agents, users must still encode their own strategies. While several types of auction protocols can be implemented, AuctionBot does not respond to changes of market conditions and is limited to price only comparison.

Kasbah is an agent-based system that bolsters both the merchant brokering and negotiation stages [1] of e-commerce. In the negotiation stage, users of Kasbah create trading agents and specify several parameters that guide their behaviors and trading actions. Some of these parameters include

1) desired date to sell an item;
2) desired price;
3) lowest acceptable price;
4) trading strategy.

For instance, a seller agent would trade between its desired (highest possible) price and lowest acceptable price. In addition, the rate at which agents increase/relax their bids/offers are determined by the strategies specified by users. Kasbah provides users with three negotiation strategies: anxious, cool-headed and frugal implemented using linear, quadratic and exponential functions, respectively. These functions govern the rate of adjusting bids and offers at each trading cycle. Although Kasbah’s agents are not restricted by allocation policies that determine which agents transact, or at what price, the strategies have fixed rate adjustments of bids/offers and do not react to changing market conditions. While Kasbah’s users manually select predefined strategies, market-driven agents autonomously select the appropriate trading strategies in response to changing market situations. As described in Section III-C, market-driven agents react to dynamics of a marketplace by making adjustable rate of concession. The series of experiments in Section IV-D suggest that the market-driven approach generally performed better than the fixed strategy approach. Similar to AuctionBot, Kasbah agents negotiate only on the price of transactions.

Tete-a-Tete provides a cooperative negotiation approach for retail markets. Unlike Kasbah and AuctionBot, where agents primarily negotiate on price alone, Tete-a-Tete’s consumer and sales agents negotiate across multiple terms including delivery times, service contracts, warranties, return policies, loan options, gift services, and other value-added services. Tete-a-Tete employs a multi-attribute utility theory (MAUT) as a decision analysis tool to model the multiple interdependent objectives that need to be considered in consumers’ decisions about sales agents’ product offerings. Shopping agents are built with software tools based on MAUT that help consumers formulate, evaluate and make value trade-off among interdependent objectives. The approach that sales agents used to configure product offerings was based on distributed constraint satisfaction. In customizing product offerings, sales agents tailored their configuration to the needs of their consumers while considering the needs of the merchants. In addition, Tete-a-Tete’s negotiation protocol is based on bilateral argumentation, comprising of exchanging proposals of product offerings from sales agents, evaluation and critiques of offerings and counterproposals. Although Tete-a-Tete’s agents are not restricted by allocation policy or predefined strategy, they are designed for cooperative negotiation in retail markets and may not be suitable for competitive situations such as stock market trading.

In CBN, agents are equipped with the ability to enhance negotiation strategies by learning from previous negotiation experience. Through the use of case-based reasoning techniques, CBN agents revise and improve negotiation strategies in each decision-making episode of the negotiation process. To learn negotiation strategies from previous experiences, CBN agents formulate negotiation strategies based on a series of offers and counteroffers. In CBN, a negotiation process consists of a number of decision-making episodes. Each episode is characterized by the evaluation of an offer, determination of strategies and generation of a counteroffer. In response to changing information including the history of the current negotiation process, agents can change their episode strategies. Agents acquire information by observing the results of negotiation in the form of offers–counteroffers as they do not know each other’s strategies. Since a series of offers and counteroffers reflect information related to episode strategies, it can be captured as a basis of the negotiation experiences. Consequently, CBN agents use episode strategies to incrementally modify offers and counteroffers toward reaching a consensus. The process of determining episode strategies using case-based reasoning consists of

1) retrieving relevant previous negotiation experiences;
2) selecting a closely matched negotiation case;
3) adapting the selected case.

Like AuctionBot and Kasbah, CBN operates on price-only negotiation. Unlike AuctionBot and Kasbah, CBN agents do not primarily rely on predefined strategies. However, CBN was designed with the assumptions that 1) the negotiation case base already contains valid and representative cases and 2) all negotiation cases provide successful negotiation experience. If no matching case is found, CBN agents select from a set of predefined strategies. In such situations, the CBN approach does not distinguish itself from the fixed strategy approach.

In Krovi et al.’s adaptive negotiation agents (ANA) [12], decision making of a negotiator is modeled with computational paradigms based on genetic algorithms. 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;
5) considering the different magnitude of initial offers.

The automatic adoption of different concession-making tactics bear some resemblance to market-driven agents. While market-driven agents respond to market situations by making different rates of concession, ANA respond to their opponents’ concession-making behaviors by adopting from among three tactics of negotiation. The three tactics include
1) reciprocating—mimicking their opponents;
2) exploiting—conceding less when opponents are more cooperative;
3) cooperating—conceding more when opponents concede more.

The feature of ANA conceding in increasing increments (e.g., 20 units in the second round, 40 units in the third round and so on) corresponds to the factor \( T - t / T \) (see Section III-B), where smaller concessions are made initially (more time remaining) and larger concessions are made in later stages of trading. At the present stage, features 2), 4), and 5) of ANA bear no resemblance to features of market-driven agents.

VI. CONCLUSION AND FUTURE WORK

The agent-based auction testbed presented in this paper introduced two distinctive features: 1) a market-driven negotiation strategy and 2) a deal optimizing auction protocol. The market-driven approach allows agents to react more intuitively to dynamically changing market environments using a simple and effective implementation of a mathematical function to automatically control and adjust the rate of relaxing transaction terms. This approach achieved generally favorable results (Section IV-D). By having a deal optimization stage, experimental results in Section IV-C showed that the deal optimization auction protocol produced generally optimistic trading outcomes (high overall user’s satisfaction). Furthermore, the experiment in Section IV-A has simulated the intuition associated with trade competitions in real life. However, as part of an on-going research initiative, further work is being done on improving the strategies of trading agents by learning the eagerness and the trading behaviors of agents. It is hoped that the trading agents and the testbed presented in this paper can shed new light on engineering agent-based auction systems.

ACKNOWLEDGMENT

The authors would like to thank three anonymous referees for their insightful comments and suggestions for improving this paper.

REFERENCES


Kwang Mong Sim received the B.Sc. degree in computer science (with honors) (summa cum laude) from the University of Ottawa, Ottawa, ON, Canada, and the M.Sc. and Ph.D. degrees in computer science from the University of Calgary, Calgary, AB, Canada, in 1990, 1992, and 1995, respectively.

Currently, he is with the Department of Information Engineering, Chinese University of Hong Kong, Shatin. His research interests include E-commerce agents, m-commerce agents, web agents, intelligent network agents, and agent theory. In addition, he reviews papers for many international journals in the area of intelligent/Internet agent. He has research/teaching experience in five countries spanning the continents of North America, Europe and Asia.

Dr. Sim was a Session Chair/Program Committee Member for the AI in E-commerce Workshops in Sydney, Australia and Melbourne, Australia, in 1999 and 2000, respectively. He was also a Session Chair/Program Committee Member for the Intelligent Agent Technology Conferences in Hong Kong and Miabashi, Japan, in 1999 and 2001, respectively. He is listed in the 17th and 18th editions of Marquis’ Who’s Who in the World.

Eric Wong received the B.Sc. degree in mathematics (with honors) from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, and the M.Sc. degree in information technology from the Hong Kong Polytechnic University, Kowloon, in 1995 and 2000, respectively.

Currently, he is with the Hong Kong Hospital Authority, Chai Wan. His research interests include agent-based e-commerce and negotiation agent.