Using Genetic Algorithms to Enable Automated Auctions

E. Wolff, M.T. Tu, and W. Lamersdorf *

Distributed Systems Group, Computer Science Department
University of Hamburg, Germany
Vogt–Kölln–Str. 30, 22527 Hamburg, Germany
[3wolff,tu,lamersd]@informatik.uni-hamburg.de

Abstract. In this paper, an approach to implement automated auctions as a negotiation mechanism for Business-to-Business electronic commerce applications is presented. It is based on Genetic Algorithms (GAs) that evolve FSMs (Finite State Machines). Each of these FSMs represents an auction strategy that competes on the market and is modified over time according to the outcome of this competition by using GA principles.
The paper gives an overview of auctions, especially in the Business-to-Business domain, and other work related to this paper. Then, the application of Genetic Algorithms to automate auctions is presented and relevant details on the prototype implementation are given. In addition, some key results obtained from experiments using this implementation are discussed.

Keywords: auctions, genetic algorithms, negotiation strategies, E-Commerce.

1 Introduction

Internet auctions are becoming more and more popular as they offer a means of doing business that has the potential to replace fixed price mechanisms of conventional commerce by those that are truly based on the *dynamic* demand for the respective goods. Presently, auction sites on the Internet mostly focus on the needs of *end-consumers*, i.e. they can be classified as a part of the Consumer-to-Consumer or Business-to-Consumer commerce. However, providers are emerging that offer auction mechanisms for the Business-to-Business domain which certainly has an enormous economic potential. With respect to this domain, possibilities of automating auctions would be of great benefit, e.g. for enterprises that belong to or depend on the supplying industry and therefore need to negotiate on many items to get best prices.

This paper presents an approach to automated auctions in the Business-to-Business domain using Genetic Algorithms. Genetic algorithms (GAs) are a

* This work is supported, in part, by grant no. La1061/1-2 from the German Research Council (Deutsche Forschungsgemeinschaft, DFG)

kind of optimization technique based on evolution principles. Due to their main feature of *self-adaptation*, negotiation strategies that are based on GAs seem to fit very well to dynamic markets in general. In [TWL00], we presented an approach to use this technique in bilateral negotiations. In this work, GAs are applied to auctions as an example of multilateral negotiations. So this work can be seen as a first step to prove the applicability of GAs to the very diverse domains in the field of E-Commerce.

The remainder of the paper is organized as follows: Section 2 gives an overview of auctions with emphasis on computer-aided or automated auctions. Then, the essentials of Genetic Algorithms, especially those based on FSMs, and how they can be applied to auctions are described in Section 3. The next section 4 gives details on the implementation of the prototype system and reports some of the results achieved from experiments done with this system. The last section 5 gives a summary and an outlook on possible future work.

2 Auctions

Generally speaking, auctions are a form of multilateral negotiations on one issue (price). Kumar and Feldman [KF98] present a classification of auctions by three key attributes:

Interaction format *Open cry auctions* are similar to auctions that take place in a public meeting. Every participant learns about every bid from the other participants and must response to a bid in a few seconds time. For internet auctions, this would mean that every participant must be online at the same time. This is almost impossible especially if the participants live in different time zones. Another problem is that this kind of auction needs a reliable distribution of the offers with low latency. Given the often unreliable and high latency connections in the Internet, it is very hard to guarantee these properties.
In a single-round *sealed bid* auction, all offers are collected until a certain deadline and afterwards evaluated. In multiple-round sealed bid auctions, there is a deadline for each round of bids and then either the auction is closed or a new round of bids is started. This kind of auctions appears to be much better suited for Internet applications because it is inherently asynchronous and thus, not all participants need to be online at the same time and offers do not need to be broadcasted with a low latency.

Control of bids and offers Either the auctioneer can announce a bid and see if one of the participants is willing to pay it or he can ask the participants to submit their bids. If the auctioneer announces the bids, he can either start with a very high bid and lower it or he can start with a low bid and raise it until only one participant is left in the auction.

Setting the trade price After the bidding phase, the bidder with the highest bids gets the auctioned item. But the price he has to pay needs not necessarily be the same as his bid. If there are multiple auctions with identical items, all winners could be allowed to pay the price of the lowest winning bid. Another alternative is to let them pay the highest non-winning bid (Vickrey auction).

Apart from these key attributes, there are many other specific characteristics of auctions such as how much information about the other offers is disclosed to each participant. The end of the auction is another example of such a characteristic: It can be closed after a certain number of rounds or at a specific point of time. Another possibility is to end the auction when the frequency of new bids decreases below a certain threshold.

For the auction system presented here, a multiple-round sealed bid auction was chosen. The auctioneer announces a minimum bid and can lower the bid if no participant is willing to pay this minimum bid. Each subsequent bid must be higher then the previous bid. The final bid is also the price the participant has to pay for the item. During the bidding process, each participant is only given the information whether his bid is currently the highest bid. The bids of the other participants are not disclosed to him. The end of the auction is reached after a certain number of auction rounds.

As mentioned above auctions are multilateral negotiations. The main motivation for implementing such an auction system is to try GAs on the rather complex field of multilateral negotiations as compared to the work already done by us in the field of bilateral negotiations [TWL00]. So this is the first step to tackle the area of multilateral negotiations using GAs.

Another reason is the rise of Business-to-Business auctions where an enterprise and its potential suppliers form a marketplace to negotiate about the price of items. In this kind of auctions, a lot of identical items are traded between two or more parties. Conducting an auction for each of these items seems impractical due to the high costs for such auctions with human participants unless there is a convenient way for the automation of these auctions. To accomplish this, there needs to be a large group of agents that take part in the auctions on behalf of the original parties. Each of these agents should then buy or sell one of the items in an auction. Note that such a technique does not depend on the fact that the price is found using an auction but could be applied to every kind of mechanism to determine a price.

3 Using FSM-based Genetic Algorithms for Auctions

To optimize groups of items, *Genetic Algorithms* (GAs) can be used. GAs are inspired by the evolution taking place in nature. This process is based on very simple principles: selection together with reproduction, crossover and mutation. Selection means that only the fittest individuals survive. Reproduction is the ability to breed new individuals and mutations are deviations during this reproduction process. Crossover is the ability to take two individuals (parents) to breed one new individual that shares some attributes with each parent.

In GAs, these basic principles of evolution are used to create objects that are optimized for a certain function. To carry out this process, a set of objects (the *population*) is evaluated at discrete points in time (between the *generations*). Each individual has a certain probability to be taken into the next generation. This probability depends on its quality (*fitness*) measured in terms of the property that should be optimized. The individual can be propagated into the next generation either unchanged (reproduction), mutated or as resulted from a crossover with another individual. The evolutionary approach can be used for the optimization of numerical problems (*Genetic Algorithms*) as well as for the automatic generation of programs (*Genetic Programming*) [BNKF98]. A benefit of these techniques is that they can be used for tasks that require the capability to adapt to a changing environment, i.e. to optimize individuals even if the criteria for the optimization change over time as in changing market places.

For auction scenarios as those described above, the optimization of a population of agents by means of Genetic Programming (GP) appears to be a good approach. An important decision in applying Genetic Programming is determining the data-structure that is used to represent the programs such as the auction strategies in this example. For the system described in this paper, Finite State Machines (FSMs) are used because they are a simple, yet enough powerful model in comparison to other approaches of applying GAs to the E-Commerce field. For example, in [Oli96] simple linear structures are used.

Usually, FSMs just accept regular sets. But in order to be applied to auctions, they must also generate some output (bids) as response to some input (state of the auction). This can be achieved by using FSMs with output such as Mealy automata [HU79]. An example of a strategy modeled as a FSM can be seen in Figure 1. After bidding the initial offer (not shown in the figure), the FSM is in the initial state (the black circle in the middle). In case the initial offer is successful — i.e. it is the highest offer in the first round — the FSM is given an input of 1 and thus continues to the right state and outputs 100 as the increase of the offer, i.e. it will increase the offer by 100 in the next round. If the initial offer was not successful, the FSM is given an input of 0 and continues to the left state, this time increasing the offer by 200. The left and right states are almost identical, the only difference is that a successful bid in the left state result in a raise of 150 while in the right state, this results in a raise of only 50. So the strategy presented here depends very much on the outcome of the initial offer: If it was successful, the FSM continues in the right state and raises the offer by relative small amounts compared to the case where the initial offer was not successful. This represents a strategy where the participant believes that in case the first offer was successful, it is likely that afterwards only relative small increases are needed to stay successful.

Note that the strategy presented here can also be given a semantics that suites the role of an auctioneer: The initial offer can be interpreted as a minimum bid and the outputs of the FSMs as amounts by which the minimum bid is decreased. So if an auctioneer follows the strategy presented in Figure 1, he will lower the minimum bid by 200 if no participant was willing to pay the initial minimum bid. In case the initial offer was successful, the minimum bid will be decreased by 100 but this will not influence the outcome of the auction as there are already participants that agreed to pay the initial minimum bid. This shows that in the system implemented, the strategies for both auctioneers and bidders are really identical; they only have different semantics depending on the role of the participant.

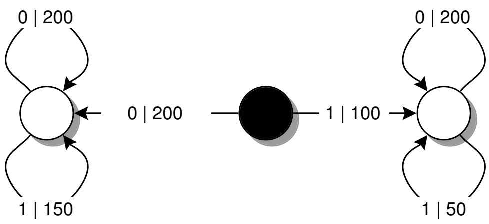

Fig. 1. A strategy for auctions as an FSM.

Besides the semantics of the FSMs, there must also be a definition of the genetic operators for crossover and mutation. Mutation on FSMs is implemented in this system by changing the states and edges. So for example, the input symbol of an edge can be modified or a state can be connected to different edges. For crossover, the sets of states of two FSMs are taken and each is divided into two subsets. Then the subsets are interchanged and edges are created as a connection between the subsets.

To calculate the fitness of a specific FSM, auctions are simulated. During each generation, every FSM takes part in at least one auction and according to the outcome of these auctions, it is assigned a fitness that determines whether it will be in the next generation or not. For the participants of the auction, each successful auction is rewarded with a fitness value of $\frac{2000}{p}$ where p is the price that the participant payed in the auction[1]. If the participant did not succeed in the auction, it is assigned a fitness value of 0 for that action. As mentioned above, FSMs can also be used to implement the strategy of auctioneers. In this case, the fitness is calculated as $\frac{p}{2000}$ because it is better for an auctioneer to achieve a higher price.

To simulate a multifaceted market, there are also *fixed strategies*. For auctions, these strategies can only choose when to increase the bid by which amount. Most often, the maximum bid the participant is willing to pay is fixed by means such as the value the participant assigned to the item. Thus, the decision how high the maximum bid can be should not be made by the strategy but rather by some external source such as the participant himself. In the MarketMaker system [MIT], a way to generate strategies for auctions is proposed: The user of the system can decide what the graph between the initial and the maximum bid should look like. For instance, one can choose to use an exponential function.

[1] The constant 2000 is chosen because the bids in the experiments done are in the order of 2000 and this fitness value will be in the order of 1.

That would mean that at the beginning of the auction, the bid is increased at lower rates than at the end of the auction. This represents a strategy where the participant wants to be cautious at the start of the auction but is willing to risk more at the end. An alternative would be to raise the bid linearly.

For the system presented here, fixed strategies with a linear increase of the bid are used. The parameters for these strategies are the amount by which the bid is raised each round and the initial bid. Both of these values are taken from a normal distribution and the mean value as well as the standard deviation of these values can be set to model a market with certain price ranges. The percentage of agents with fixed strategies to take part in each auction can be set and so a complete market with a certain demand and certain prices can be simulated. Again, the same kind of strategies can be used for auctioneers, so apart from the demand, also the supply of the market can be simulated using these fixed strategies.

4 Implementation and Results

To implement the concepts presented in the last section, an object-oriented approach was chosen and the Java programming language was used. In this section, the prototype auction system based on GAs will be presented. Then, some experiments done using this prototype system and their results will be described.

The prototype auction system consists of a part that is responsible for auctions in general. Within this part, it is defined how the auction should be carried out and the functionality to implement bidders and auctioneers with fixed strategies is implemented. However, the implementation of the auction itself makes no assumptions about the bidders and auctioneers and thus, FSMs can easily take part in these auctions. Every time an auction takes place, not only the bidders are assigned a fitness but also the auctioneer. Both the auctioneers and the bidders can be FSMs that are optimized by a Genetic Algorithm. Most of the functionality for general negotiations and GAs was provided by a framework for the implementation of Genetic Algorithms that was built at the University of Hamburg to study GAs in the context of negotiations and auctions (see [TWL00] for more details on the framework). After this system was implemented, a number of experiments, also called *scenarios*, were performed and results were measured. These scenarios share the parameters shown in Table 1.

Each scenario was run with values for the mutation and crossover probability in the range from 0 to 0.5 in steps of 0.05. In the 20th generation, the prices for the bidder with strategies optimized by a GA and for the bidders with fixed strategies was measured. In addition, the ratio between auctions where a fixed strategy succeeded and auctions where a strategy optimized by a GA succeeded was measured. The same was done for the strategies of the auctioneers. For the auctioneers, the chosen parameters for both the fixed and the genetically optimized strategies can be found in Table 3 while the parameters for the bidders can be found in Table 2.

parameter	value
size of population	20
generations	20
iterations (for mean values)	100
auctions per generation	4
rounds per auction	5

Table 1. Common parameters for all scenarios.

	genetically optimized strategies	fixed strategies
participant per auction	2	50 %=2
initial offer	0-2000	$N(900, 100)$
increase per round	0-120	$N(100, 10)$
	FSM have $\leq$ 10 states initially.	linear increase

Table 2. Parameters for the bidders. $N(x, y)$ denotes a normal distribution with mean value x and standard deviation y.

The first relevant result achieved is that with respect to the auctioneer, both fixed strategies and strategies optimized by a genetic algorithm perform equally well in all scenarios. This is not surprising since auction is a market type where bidders determine the price by outbidding each other. What the auctioneer does during the auction can not influence this.

For the bidders, first a scenario was implemented where in each auction, there were two strategies that were optimized by a Genetic Algorithm. This resulted in a competition in which these strategies tried to outbid each other. The reason for this is that the strategy that succeeds in an auction is assigned a fitness > 0 while the strategy that does not succeed is assigned a fitness of 0. So an evolution takes places where the strategies with lower maximum bids are slowly extinguished because of their lower fitness values. At the same time, the genetically optimized strategies succeed in virtually every auction. So there is a close connection between the maximum offers and the rate of success in the auction: If the maximum bid is very high, one can succeed in almost all auctions. This translates to a high market share in the market of the auctioned items.

	genetically optimized strategies	fixed strategies
initial minimum offer	0-2000	$N(900, 100)$
decrease per round	0-100	$N(100, 10)$
	FSM have $\leq$ 10 states initially.	linear decrease

Table 3. Parameters for the auctioneers.

So for the next scenario, only one strategy that is optimized by a Genetic Algorithm is allowed in each auction. In that case, the market share of the strategies optimized by a genetic drops to about 80 %-90 % and the prices paid by them also drop, but they are still higher than the prices for fixed strategies. Again, this is not surprising because higher maximum bids lead to higher market shares as stated above.

The real problem of these scenarios is that the implemented system has not taken into account the actual goal or the value function of the person or company that wants to use the strategies. Two simple goals can be considered: purchase as many items as possible for a certain total price *or* purchase a certain number of items for a price as low as possible. A certain number of purchased items translates to a certain number of purchased items. For example, if the market share is 40 % and there are 1000 auctions, then 400 items are bought. Once the market share and the price can be controlled, more complex value functions representing a relationship between price and market share can be specified by the user. Therefore, in the next scenario, the strategies were assigned a fitness > 0 only if they bought the item for a price *below* a certain threshold, and in fact then, the strategies are really optimized by the Genetic Algorithm to fulfill this requirement because the evolution leads to a population of strategies with a mean price well below this threshold. To control the *market share*, a scenario was implemented in which the strategies were assigned a fitness of 0 if they purchased more items than specified by the user. This led to a population that did not exceed the specified market share. However, at the same time, the fixed strategies achieved a higher market share with lower bids. So even though the goal was reached by the population, they performed worse than the fixed strategies. At least, it was shown that GAs can fulfill the goals of a user.

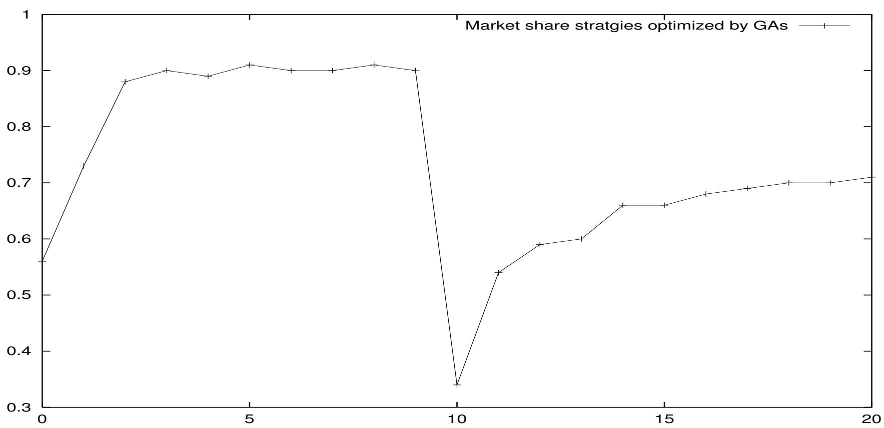

Fig. 2. Market share for the strategies optimized by Genetic Algorithms in each generation.

Another important point to take into account is the question how fast the population can adapt to changes in the market. To answer this question, the values for the fixed strategies were changed in the 10th generation: Instead of an initial offer of 900, the fixed strategies now make an initial offer of 1500. This means they are willing to pay 600 more for the items and thus a change in the marketplace is simulated. Also the number of auctions per generation was lowered to two. The changes for the market shares and the prices are given in Figures 2 and 3. From the first generation, it takes 3 generations (6 auctions) until the market share reaches a value that stays constant until generation 10. At that point, the market share drops because of the price increase. This time, it takes 4 generations (8 auctions) until a stable value is reached again. This gives some hints on how long the results of the strategy population stay sub-optimal if a change occurs in the market. However, it is important to note that in real markets, the changes are usually not as dramatic as the changes presented here. So in a slowly changing environment, the Genetic Algorithms could perform even better than the values presented here suggest.

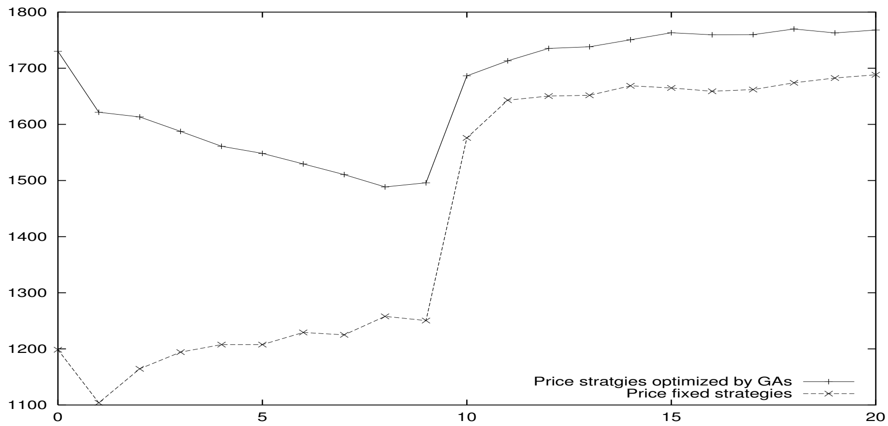

Fig. 3. Prices for the fixed strategies and the strategies optimized by a Genetic Algorithm in each generation.

5 Summary and Outlook

In this paper, a method to make a population of agents learn strategies for multilateral negotiations (i.e. auctions) in Business-to-Business E-Commerce based on Genetic Algorithms (GAs) has been presented. To model the strategies, Finite State Machines (FSMs) are used as the basic data structure processed by the GAs.

To explore the applicability of GAs as a negotiation mechanism, especially as strategies for auctions, a corresponding prototype auction system was built and several experimental scenarios were implemented. The first scenarios lacked the possibility for the user of the strategies to specify his preferences. It was realized that the most important values for the auctions are the maximum price one is willing to pay and themarket share he wants to reach. In the scenarios implemented later, the user could specify the market share he wants to achieve. This is closely related to the number of items the user wants to purchase. The other possibility is to specify the maximum price the user is willing to pay for the items. In both cases, the strategies behaved as specified by the user. However, at least in a few cases, fixed strategies are still more successful, i.e. reach a higher market share while paying lower prices. Other experiments showed that the population is able to adapt quite quick to changes in the market.

These results show that Genetic Algorithms might provide good strategies for automated negotiations — which can be performed by mobile agents for example (see [TGML98]) — in E-Commerce systems in cases where larger quantities of items are traded in regular time intervals. Auctions are only one way of implementing such systems; other mechanisms to agree on a price could also be tackled by this technique. The main benefit is the automatic adaption to changes in the market without human interference.

Several improvements can be still be achieved w.r.t. the Genetic Algorithms used in the system described here, as only very basic algorithms have been used now and could be easily replaced by more sophisticated GAs. Also other data structures than FSMs could be used. The main conclusion that can be drawn from the presented work so far is that Genetic Algorithms apparently match the requirements for enabling automatic Business-to-Business E-Commerce quite well and therefore, further research in this area might be very fruitful.

References

[BNKF98] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming — An Introduction: On the Automatic Evolution of Computer Programs and its Applications.* Morgan Kaufmann Publishers, 1998.

[HU79] J. E. Hopcroft and J. D. Ullman. *Intrduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[KF98] Manoj Kumar and Stuart I. Feldman. Business negotiations on the internet. Technical report, IBM T.J. Watson Research Center, 1998.

[MIT] MIT Media Lab. Marketmaker. http://maker.media.mit.edu/.

[Oli96] J. R. Oliver. *On Artificial Agents for Negotiation in Electronic Commerce.* PhD thesis, The Wharton School, University of Pennsylvania, 1996.

[TGML98] M.T. Tu, F. Griffel, M. Merz, and W. Lamersdorf. A Plug-In Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents. In K. Rothermel and F. Hohl, editors, *Proc. 2. Intl. Workshop on Mobile Agents, MA'98, Stuttgart*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[TWL00] M.T. Tu, E. Wolff, and W. Lamersdorf. Genetic Algorithms for Automated Negotiations: A FSM-based Application Approach. In *Proceedings of the 11th International Conference on Database and Expert Systems Applications (DEXA 2000) — e-Negotiations Workshop*, 2000.