Effects of Tree Size and State Number on GP-Automata Bidding Strategies

Charles W. Richter, Jr.
Electrical and Computer Eng. Dept.
Iowa State University
Ames, Iowa 50011
crichter@iastate.edu
(515) 294-4597

Daniel Ashlock
Mathematics Department
Iowa State University
Ames, Iowa 50011
danwell@iastate.edu
(515) 294-7965

Gerald Sheblé
Electrical and Computer Eng. Dept.
Iowa State University
Ames, Iowa 50011
gsheble@iastate.edu
(515) 294-3046

ABSTRACT

The impending deregulation of the electrical industry in the USA promises to open a multi-billion dollar industry to competition. Current research indicates that the double auction will be at the heart of several regional electrical commodity exchanges. The authors are attempting to design comprehensive profitable bidding strategies for traders. The advantages of the strategies detailed here come from using data structures which combine genetic programming and finite state automata termed *GP-Automata*. Adaptive strategies encoded by two populations of GP-Automata are tested in an auction simulator modeling distribution companies and generation companies buying and selling power via a double auction. In addition to evolving profitable bidding strategies, the resulting strategies can also be designed to imitate certain types of trading behaviors. These strategies can be used directly in on-line trading, or as realistic competitors in an off-line trading simulator. In this paper we report the results of specific experiments which test the effect of changing the size of the GP trees, and the effect of changing the number of states.

1. Introduction

Regulations governing the electric utility industry in the United States are being changed to promote competition. By increasing competition through deregulation of the electrical transmission network, the Federal Energy Regulatory Commission (FERC) hopes to increase power system efficiencies and to see benefits for electric consumers.

For decades, electric consumers in the US had only their local vertically integrated utility as a source of electricity. Utilities have always been guaranteed customers, and have not had to operate in a competitive environment. Both consumers and generators of electricity (or their representatives) will soon be faced with having to sell or purchase power through a commodity exchange. To be successful, these electricity traders will need to develop bidding strategies. This paper focuses on developing such strategies.

The research presented in this paper assumes an electric marketplace which is structured similar to commodities exchanges like the Chicago Mercantile Exchange, Chicago Board of Trade, and New York Mercantile Exchange (NYMEX) where commodities (other than electricity) have been traded for many years. The fact that in 1996, NYMEX actually added electricity futures to their offerings supports the authors' predictions outlined in previous papers [12,17,18,19] regarding the coming competitive environment.

In our research, trading agents use a genetic algorithm (GA) to evolve bidding strategies for the electric energy market. These strategies are coded in the form of finite automata coupled with genetic programming (GP-Automata) [2,3]. An optimal bidding strategy must be adaptive, able to properly react as the trading behavior of its competitors changes. Coding information in the form of GP-Automata, which evolve in a GA, allows complex adaptive strategies to develop. The results have been written up specifically for the electric energy market, but are directly applicable for other markets.

Part 2 of this paper surveys recently published work in this area, including research on evolving economic agents, genetic programming applied to auctions, and auction environments. Part 3 describes GP-Automata and the auction environments in which the strategies are designed to perform. Part 4 describes the experiment performed to test the effect of modifying the tree size or the number of states in the GP-Automata, and presents the results of these experiments. Part 5 presents the conclusions made as a result of the research and Part 6 lists some possible directions in which this research can be extended.

2. Review of Recent Work

Some research has been done in developing bidding strategies for electric systems in other countries. Finlay [6] analyzed bidding strategies for the restructured Power Pool of England and Wales system, and showed mathematically that there exists an optimal bidding strategy for its bidders. Finlay's work differs in that his objective was not to maximize the profit of the individual generation companies, and the system itself is different from those proposed in the USA. Hence it is not directly applicable to our scenario.

Sheblé [19] described the different types of commodity markets and their operation and outlined how each could be applied in the evolved electric energy marketplace. Under the framework described by Sheblé [18], companies presently having both generation and distribution facilities would, at a minimum, be divided into separate profit and loss centers. Power is generated by generation companies (gencos), transported via transmission companies (transcos), and all power is sold to distribution companies (distcos).

The framework described by Sheblé [18] allows for a cash market, a futures market and a planning market. The cash market is for trading power for each 30 minute period in the next 30 days. The futures market allows electricity trading from 1 to 18 months into the future. Futures contracts are non-firm for a specific month. This futures market provides a means for electricity traders to manage their risk. The other market is a planning market that can be used to develop capital to build new plants and would allow trading more than 18 months into the future. Fig.1 shows how these markets are interconnected. Sheblé et al. [20] outline how cash, future, planning and swap markets can handle real-time control of the system (e.g., automatic generation control) and risk management.

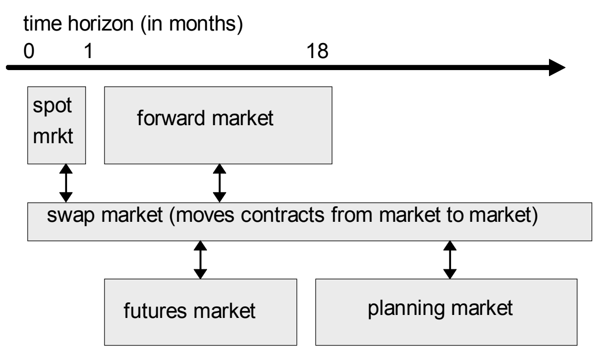


Fig. 1. Interconnection of the various markets.

Work by Kumar and Sheblé [11] brought the above ideas together and demonstrated a power system auction game designed to be a training tool. That game used the double auction mechanism in combination with classical optimization techniques. Buyers and sellers interact through a central coordinator, an Independent Contract Administrator (ICA), who matches the bids subject to all operational constraints. The central coordinator is responsible for ensuring that the energy transactions resulting from the matched bids do not overload or render the electrical transmission system insecure. Gencos and distcos coordinate only via the prices transmitted to a central auctioneer. The ICA monitors and responds to the power system limits and transmission capacities.

Developing bidding strategies with evolving trading agents for the deregulated electrical utility industry is a new field of research. Apart from the electrical utility industry, interest has grown in recent years for using evolving, or adaptive, agents to simulate trading behavior. Research with adaptive agents has proved to be a useful means of exploring trading markets outside of the electrical industry.

LeBaron [13] uses evolving agents to learn to play financial markets. Tesfatsion [24] describes research in which trading agents decide who to trade with based on an expected payoff. Ashlock [2] uses genetic programming combined with a finite state automata to play a classic academic game called Divide the Dollar which involves bidding behavior and strategies. Ashlock and Richter [3] use the same game to study kinship effects and conclude that when evolving buyers and sellers, unless they come from separate populations, collusion is likely to occur.

Andrews and Prager [1] used a game based on a double auction to verify that genetic search is useful. They show that GP-based agents actually do learn and they compare the performance of the GP-based strategies to those developed using simulated annealing. In addition, they show that at the beginning of the genetic algorithm it is possible to use a less rigorous fitness test than needed in later generations. While their findings may be useful to the genetic algorithm community, their experiments leave room for further improvements in strategy-building.

3. Methods And Techniques

This section outlines the methods that were used to simulate the marketplace; it introduces the basics of genetic algorithms, genetic programs and GP-Automata; and goes through the process of developing a bid for an auction from a given GP-Automaton.

3.1. The Marketplace

As in [17], the authors again assume the existence of regional commodity exchanges in which buyers and sellers participate in a double auction. This framework has been adopted from Sheblé [18], which is an extension of the framework being implemented in California. For the results presented in this paper, transcos are considered to be exogenous to the market, distcos and gencos are allowed to interact in an environment as described in the previous section. Although our framework covers the futures and options markets, the research described in this paper is written up for only the cash / spot market.

In the double auction used for this research, the bids and offers are sorted into descending and ascending order respectively, similar to the Florida Coordination Group approach as described by Wood and Wollenberg [26]. If the buyer's bid is higher than the seller's offer to be matched, then this is a potentially valid match. The ICA must determine whether the transaction would compromise system security and whether sufficient transmission capacity exists. If the ICA approves, each potentially valid offer and bid helps to determine the final price, termed the *equilibrium price*. The midpoint price of each pair (weighted by the number of MWs) is used to determine one overall equilibrium price. This slightly different from the power pool split savings approach that many regions have been using for years in that each of the valid contracts has the same price. An example is given in Table 1.

Table 1. Example of auction bid matching.

Buy bids ($/MW)	Sell offers ($/MW)	Contract ?	Midpoint of bid and offer	Equilibrium price ($/MW)
12.50	8.50	Yes	10.50	10.63
12.00	9.00	Yes	10.50	10.63
11.80	10.00	Yes	10.90	10.63
10.00	10.50	No	NA	NA
9.50	11.00	No	NA	NA

In the example shown in the table, there are three bids that are higher than the corresponding offers. If there is not a sufficient number of valid matches, then *price discovery* has not occurred. The auctioneer reports the results of the auction to the market participants. When all bids and offers are collected and insufficient valid bids and offers are found to exist, the auction has gone through one cycle. The auctioneer then reports that price discovery did not occur, and will ask for bids and offers again. The buyers may increase their bids and sellers may decrease their offers and another cycle of the auction is played. The cycles continue until price discovery occurs, or until the auctioneer decides to match whatever valid matches exist and continue to the next round or hour of bidding.

After price discovery, the auctioneer asks if another round of bidding is requested. If the market participants have more power to sell or buy, they request another round. Allowing multiple rounds of bidding each time period allows the participants the opportunity to use the latest pricing information in forming their present bid. This process is continued until no more requests are received or until the auctioneer decides that enough rounds have taken place. See Fig. 2 for a block diagram of the auction process.

3.2. The Basics of Genetic Algorithms

Derived from the biological model of evolution, genetic algorithms (GAs) operate on the Darwinian principle of natural selection [7]. A population of data structures appropriate for the optimization problem are "randomly" initialized. Each of these candidate solutions is termed an individual or a creature. Each creature is assigned a fitness, which is simply a heuristic measure of its quality. Then during the evolutionary process, those creatures that have a higher fitness are favored and allowed to procreate.

During each generation of the evolutionary process, creatures are randomly selected for reproduction with some bias toward higher fitness. After parents are selected for reproduction, they produce children via the processes of *crossover* and *mutation*. The creatures formed during reproduction explore different areas of the solution space than did the parents. These new creatures replace lesser fit creatures of the existing population. The basic algorithm can be written as follows:

1. Randomly initialize a population and set the generation counter to zero.
2. Until done or out of time, do the following:
   - Calculate the fitness of each member of the population.
   - Select parents using some fitness bias.
   - Crossover the parents to create candidate offspring.
   - Mutate these new offspring.
   - Replace the lesser fit members with the offspring.

Increment the generation counter and go to step 2.

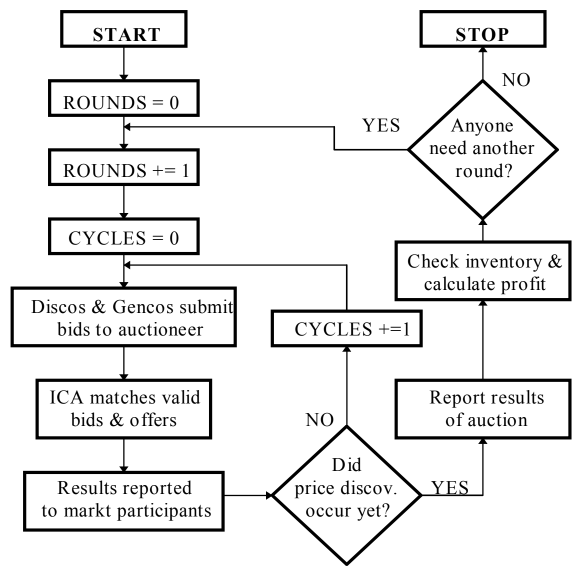


Fig. 2. The auction process.

The parents are required to be in pairs for reproduction, and the result is two children. Children are created by copying the contents of parent 1 into child 1 and of parent 2 into child 2 until a randomly selected crossover location is reached. At this point, bits are copied from parent 1 into child 2, and from parent 2 into child 1.

Following the crossover process, the children are mutated. Mutation introduces new genetic material into the gene at some low rate. If the gene to be mutated in the child is represented by a string, assuming a binary string mutation involves flipping the bit (0 goes to 1, 1 goes to 0) at each location in the string with some probability. If the gene is represented by an integer, mutation might involve adding an integer that will result in a different valid integer occupying that gene location (loci).

3.3. The Basics of Genetic Programming

The process of genetic programming has been called automatic programming and is a sub-class of the genetic algorithm field. Genetic programming is a fairly new discipline and is attributed to John Koza [9]. Typically shown in either parse tree (see Fig 3.), or S-expression form (e.g.: split(ite(sub(hbb, cost), 20, asb), 10)). Genetic programs (GPs) are evolvable programs. Each parse tree contains some number of nodes and branches. The branches connect the various nodes which can be either an *operational node* which has arguments and performs some operation involving those arguments, or *a terminal node* which returns a constant value.

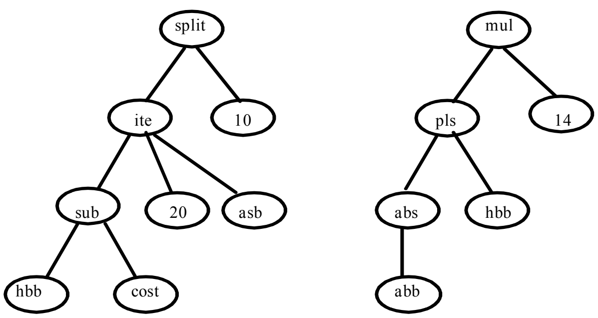


Figure 3. Sample GPs.

The designer specifies the set of valid operators and terminals suitable to the problem being investigated. For instance, in developing bidding strategies, suitable operators and terminals might be those described in Table 2. In designing GPs for the GP-Automata, it is desirable to give the trees an opportunity to return numbers in the range of competitive bids.

The GPs are traditionally evolved in a standard genetic algorithm (as described in the previous section) with the following modifications. Crossing over two parents involves randomly selecting a node from each parent and swapping the sub-trees rooted at those nodes. *Mutation* involves randomly selecting a node in the candidate child and throwing away its sub-tree. In its place a new sub-tree is generated randomly. See Koza [9,10] for a more detailed look at GP.

Table 2. Valid operators and terminals for the GP.

Name	Type	Args	Description
gte	oper	2	Return 1 if 1st arg is >= 2nd arg; otherwise return 0.
ltn	oper	2	Return 1 if 1st arg is <= 2nd arg; otherwise return 0.
ite	oper	3	If 1st arg is even after truncation, return 2nd arg; else, 3rd arg.
abs	oper	1	Returns absolute value of arg.
mid	oper	2	Returns average of the two args.
mul	oper	2	Returns multiplication of 2 args
pls	oper	2	Returns addition of 2 args
sub	oper	2	Subtracts 2nd arg from 1st arg
max	oper	2	Returns the larger of the two args
min	oper	2	Returns the smaller of the 2 args
cyc	term	0	Returns current cycle number
cst	term	0	Returns the cost of gen. for the bid
asb	term	0	Returns the average sell bid
hsb	term	0	Returns the max sell bid
lsb	term	0	Returns the min sell bid
abb	term	0	Returns the average buy bid
hbb	term	0	Returns the max buy bid
lbb	term	0	Returns the min buy bid

3.4. GP-Automata

GP-Automata combine finite state automata with GPs. They were first described as such by Ashlock [2] and were used by Ashlock and Richter [3]. The typical finite state automaton specifies an action and "next state' transition for a given input or inputs. With only one or two binary inputs to work with, it can be fairly simple to develop a finite state diagram to cover the possible input/output relations. When the number of inputs is large the task is much harder. The number of transitions needed to cover all possible combinations of inputs grows exponentially (e.g., 10 inputs each having 5 possible values would require 5.0^{10} transitions). This is where genetic programming comes in. The GP-trees are bandwidth compressors. They are used by the GP-Automata for selecting which inputs to consider and for performing computations involving these inputs. See Fig. 4 for an example of a GP-Automaton.

	IF ODD		IF EVEN		
State	Action	Next State	Actio n	Next State	GP (Decider)
1	14.5	1	U	1	Ite (mul(10, abs (hbb))
2	*	1	37	3	ite(max(10, asb), hbb, lbb)
3	12	2	5	1	split (5, abb)
4	U	3	*	2	47
Initial Action	24	Initial State	2		

Fig. 4. A four state GP-Automaton.

Reading the rule encoded in the above GP-Automaton is fairly simple. We see that this automaton begins by bidding the number in the 'initial action' field. Following the initial action, the 'initial state' tells us which state we would use next (in this case, 2). The GP-Automaton in the figure has four states. Coupled with each of these states is a GP-tree termed a decider. When executed, the decider returns a value between 0-100. Based on that returned value, one of the following two things will happen: (a) if that value is *even* after truncation, the action listed under 'IF EVEN' is taken and we move to the next state listed under 'IF EVEN'; (b) if the returned value is *odd* after truncation, then we use the action and next state listed under 'IF ODD'. The 'action' is the number listed in the action field of the automaton, with two exceptions. The first exception is the 'U' which indicates that the value returned by the decider should be taken directly as the action. The second exception is a '*' which indicates that further computation is necessary and hence the GP-Automata refrains from acting immediately. Instead, it immediately moves to the next state. This gives rise to the possibility of complex (multi-state) computation as well as infinite loops. To prevent infinite loops, after an externally specified maximum number of '*'s, an action is selected at random from actions uniformly distributed over the valid range.

In this paper we evolve a population of GP-Automata in a GA. After selecting parents as described in Part 4, offspring are produced using crossover and mutation. Crossover for the GP-Automata involves selecting (with a uniform probability) a crossover point ranging from zero to the number of states. We then copy parent1's states from zero to the crossover point to child1 and parent2's states to child2. Following the crossover point, child1 gets parent2's state information and child2 gets parent1's state information (including the associated decider). Before replacing less fit members of the population, each child is subjected to one of four types of mutation. MutationA is standard GA mutation which selects a state or action at random and replaces it with a valid entry. MutationB selects two states within a candidate child and swaps the deciders (intact) associated with these states. MutationC performs the GP crossover as described previously on two states selected randomly from the candidate child. MutationD generates an entirely new decider for a randomly selected state within the candidate child.

3.5. Using the Strategies in an Auction

In each generation, the performance of each GP-Automata in the population is tested by using the strategy in an auction. GP-Automata representing seller strategies from the first population attempt to contract with GP-Automata representing buyer stratgies from the second population. The price and the resulting profit is a function of all bids and offers, not just the bids produced by an individual GP-Automaton. Randomly assigning members of the population so that they aren't always paired with the same group of buyers and sellers helps to ensure that the resulting GP-Automata strategies will be robust. At the end of the competition, some fitness measure (e.g., profit from the contracts) is assigned to each GP-Automaton.

During the first cycle of bidding in an auction, the strategy defined by the GP-Automata uses the bid specified in the 'Initial Action' cell and goes to the state listed in 'initial state'. For each subsequent cycle of bidding the results of the previous cycle of bidding are available as inputs to the GPs. These inputs are supplied by the terminals listed in Table 2. The GP-trees use the information stored in the terminals as well as numerical terminals in the range 0-100. Bids are taken from the action cell of the automata, except in the cases where the action is listed as a '*' or a 'U', as described previously. The bids are submitted, along with the bids from the competing sellers and buyers, to the auctioneer for evaluation. The bids and offers are matched and a would-be price is reported, completing one cycle of the auction. The cycles continue until price discovery occurs or until some maximum number of cycles (maxcycles) has passed. There is a maxcycles parameter, which is selected uniformly over a range to prevent the strategies from falling into a local optima in which the strategies work well when the number of cycles is identical over the trials in a given generation.

4. Experiment Design and Results

Past experiments with Divide the Dollar [2] indicated that the GP-Automata would be well suited for developing bidding strategies for multiple buyer, multiple seller auctions, like those described in part 3 of this paper. Therefore, many of the parameters used here are set fairly closely to parameters used in the previous experiment. For the control case, each of the GP-Automata was allowed six states/deciders. They were allowed to have GP-trees with no more than 16 nodes. We used tournament selection with tournament size four and a population size of 24. The deciders were allowed to use those terminals and operators described in Table 2.

The double auction in which the strategies were tested matches 5 buyers and 5 sellers. While the GP-Automata are being tested for fitness, they compete in an auction: 5 members of their own population, against 5 members of the other population (not necessarily distinct). Since each strategy's performance depends on who it is competing with, buyers and sellers are arranged so that in each column, all buyers or sellers will appear exactly once. An example is given in Table 3. This is not an exhaustive testing of all groupings, but rather an indicative sample. After selecting a grouping (row) of sellers, the buyers are assigned as shown on the right side of the table. That row of sellers competes against each grouping of buyers once. Each GP-Automaton is assigned fitness in proportion to the profit it made from resulting contracts. At the next grouping of sellers, the buyers are reassigned in the same manner, and the process continues.

To put the GP-Automata on equal footing, each buyer receives the same amount from his customers (which corresponds to 75 on the graphs shown) for the electricity it buys, and each seller is able to produce electricity at the same cost (corresponding to 25 on the graphs shown). Profit is calculated in the usual way. For the buyers, profit is the equilibrium price minus the cost of production, multiplied by the contract amount. For the sellers, profit is the revenue received from their customers minus the equilibrium price, multiplied by the contract amount. (**Note**: These costs and revenues, as well as the bids, are show ranging from 0-100 in the graphs in this paper. In order for the numbers to be realistic they must be scaled linearly such that 100 corresponds to \$20/MW, and 50 corresponds to \$10/MW, etc.)

Table 3. Assignment of buyers & sellers to an auction.

Sellers					Buyers				
21	19	4	3	21	23	14	20	5	9
4	11	2	7	16	7	24	13	1	3
3	22	16	24	17	10	11	8	6	23
…	…	…	…	…	…	…	…	…	…
12	9	15	8	5	19	1	3	17	1

Although more cases were actually run, three cases are presented in the results section of this paper. The first is a control case, the results of which are shown in Figs. 5 & 6. The control case has a population size of 24 GP-Automata for both the buyers and the sellers. Each GP-Automaton has six states. Each decider in the GP-Automata is limited to a maximum of 16 nodes. The max, min, and avg. fitness and a histogram of bids averaged across all 40 runs is show in Fig. 5 (a, b, & c). A typical run from control case is shown in Fig. 6 (a, b, & c). The histogram shows the number of bids that occurred within a particular range (0-100 in discrete intervals of 5) at each generation.

The second case limited the sellers to having a maximum of one node in their deciders. The buyers are as described in the control case. Fig. 7 (a, b, & c) shows the results averaged across 40 runs for case 2. Averaged across the 40 runs, the maximum, minimum and average fitness are very similar to the control case. The histogram shows that both populations of GP-Automata were somewhat more exploratory in nature having a wider distribution of bids than in the control case. Fig. 8 (a, b, & c) shows the results from a typical run of case 2.

The third case presented in this paper limits the number of states that the sellers can have to one. The deciders are at a maximum of 16 nodes. The buyers are as described in the control case. Fig. 9 (a, b, & c) shows the results averaged across 40 runs for case 3. Again the fitnesses were very similar to the control case. The histogram shows that the bids were distributed in much the same way when averaged across all runs, but there is a tendency for them to be less exploratory within a particular run (See Fig. 10 (a, b, & c). This shows up in Figure 9 (c) in that the bids are in straight rows over the generations.

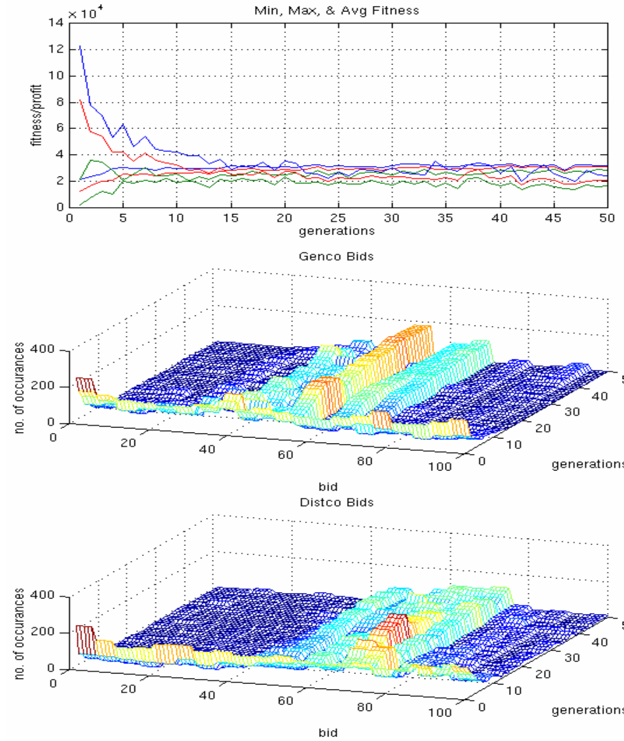


Fig. 5 (a, b, c). Results for the control case.

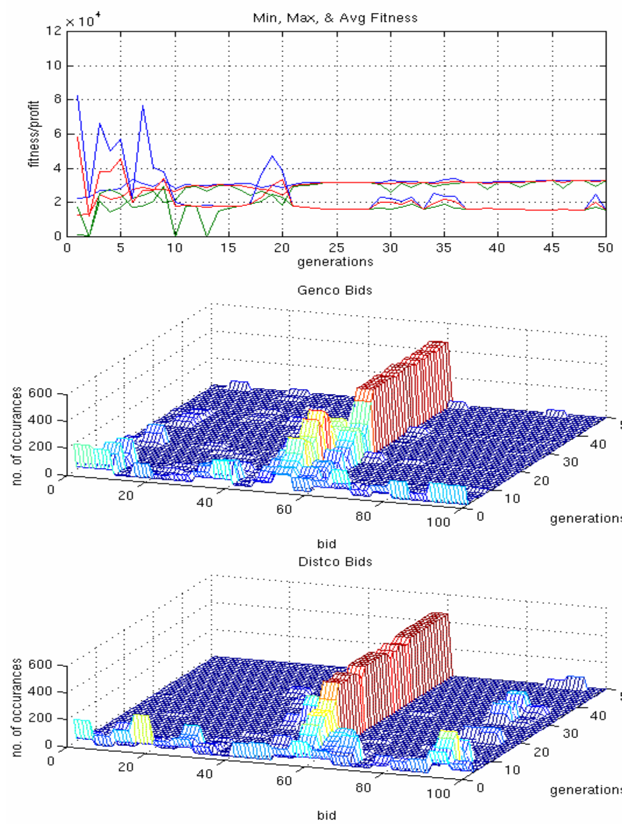
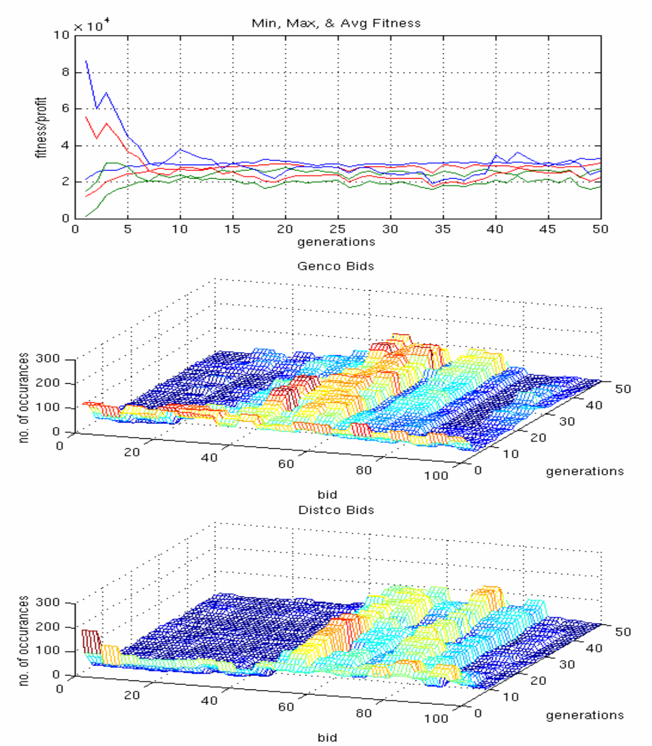
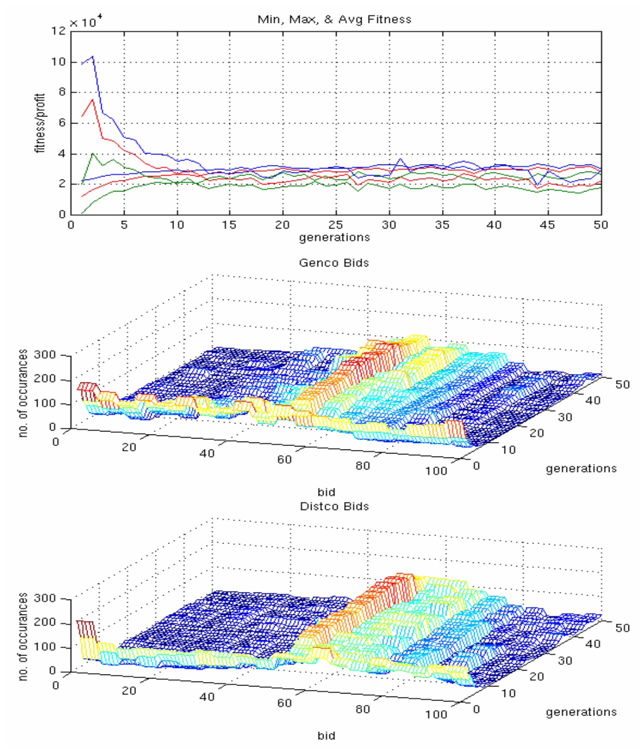


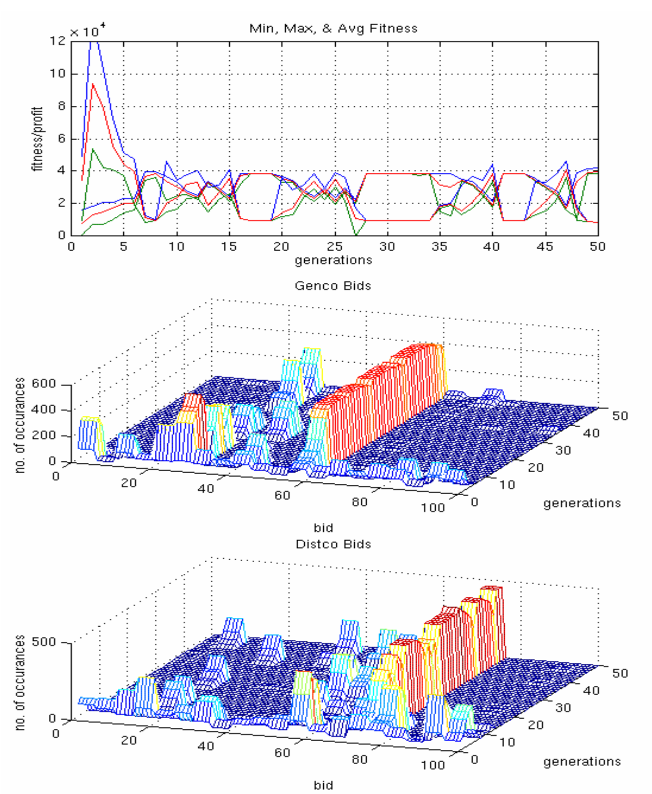
Fig. 6 (a,b,c). A typical run from the control cases.



Fix. 7 (a,b,c). Average across 40 runs. Gencos are limited to one node GPs.



Fix. 9 (a,b,c). Average across 40 runs. Gencos are limited to having only one state.



Fix. 8. A typical run from the on node GP Genco case.

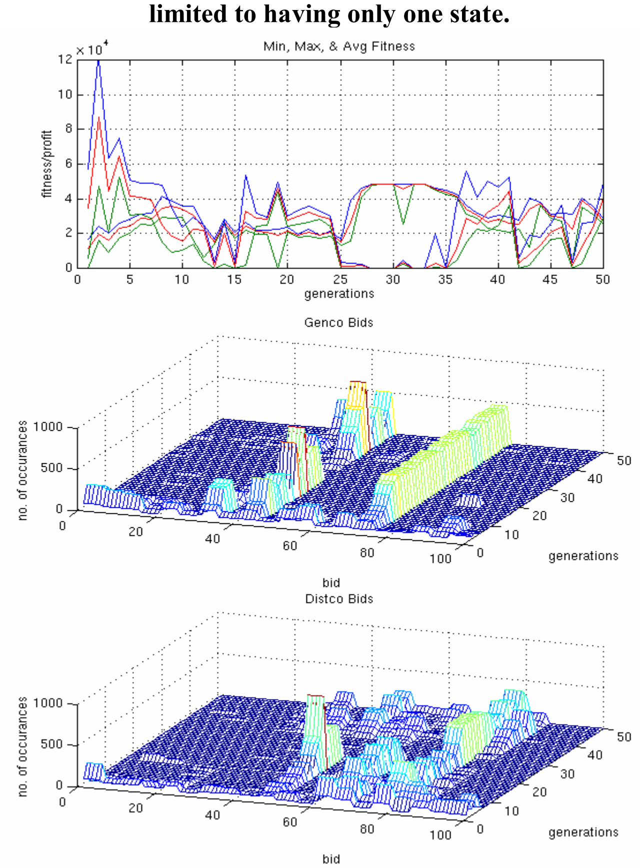


Fig. 10 (a,b,c). Typical run for the single state Genco case.

5. Conclusions

The results demonstrate that when GP-Automata bid in a multi-participant auction, they learn to bid in a sensible and explicable manner. The GP-Automata lend themselves well to scenarios where there are vast amounts of data available, and identification of crucial data is important. The company models used in the simulations described in this paper were fairly straight forward. Adding more detail (e.g. ATCs, forecasted prices, unit commitment schedules) will increase the volume of information that the bidder needs to consider in making a bid. The fact that GP-Automata suffer very little degradation in performance when they are limited by number of states or by tree size gives some indication of how powerful the method is.

6. Future Research

The auction that we have used is more realistic for the coming deregulated electricity marketplace than in [1,2,3], but more details remain to be added which will complicate matters. We plan to test in further experiments that the GP-Automata are able to make use of this additional data to increase the performance of the strategies. As Fig. 1 shows, traders have to deal in more than one market. Extending the strategies to cover multiple markets is another area that the authors are currently pursuing.

In addition to adding the above details, we need to perform a more complete sensitivity analysis to see how the various parameters affect performance of the strategies that are constructed. Among these parameters are the parent selection methodology, the population size, and modifications on the auction.

BIBLIOGRAPHY

[1] M. Andrews and R. Prager, "Genetic Programming for the Acquisition of Double Auction Market Strategies." *Advances in Genetic Programming*. Edited by K. Kinnear, Jr. Cambridge, Massachusetts: The MIT Press, 1994.

[2] D. Ashlock, "GP-automata for Dividing the Dollar." Mathematics Department, Iowa State University, Ames, IA 1995.

[3] D. Ashlock and C. Richter, "The Effect of Splitting Populations on Bidding Strategies." GP-97 Proceedings of the Second Annual Conference, pp. 27-34. San Francisco, CA: Morgan Kaufmann Publishers.

[4] B. Barkovich, and D. Hawk, "Charting a new course in California" , *IEEE Spectrum*, Vol. 33, July 1996, p26.

[5] G. Fahd, and G. Sheblé, "Optimal power flow emulation of interchange brokerage systems using linear programming." *IEEE Transactions on Power Systems*, Vol. 7, No. 2, pp. 497-504, 1992.

[6] D. Finlay, *Optimal Bidding Strategies in Competitive Electric Power Pools*. Masters thesis, University of Illinois, Urbana-Champaign, IL, 1995.

[7] D. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1989.

[8] Y. Ioannides, "Evolution of Trading Structures." Department of Economics, Tufts University, Medford, MA 02155, 1995.

[9] J. Koza. *Genetic Programming*. Cambridge, Massachusetts: The MIT Press, 1992.

[10] J. Koza. *Genetic Programming II*. Cambridge, Massachusetts: The MIT Press, 1992.

[11] J. Kumar, G. Sheblé, "Auction Game in Electric Power Market Place." *Proceedings of the 58th American Power Conference*, 1996, pp. 356-364.

[12] J. Kumar, G. Sheblé, "Framework for Energy Brokerage System with Reserve Margin and Transmission Losses." 1996 IEEE/PES Winter Meeting, 96 WM 190-9 PWRS. New York: IEEE

[13] B. LeBaron, "Experiments in Evolutionary Finance." University of Wisconsin -Madison, Madison, Wisconsin, 1995.

[14] P. Milgrom, *Auctions and Bidding: A Primer*. Journal of Economic Perspectives, Vol. 3, No. 3. Summer 1989, pp. 3-22, 1989.

[15] D. Post, *Electric power interchange transaction analysis and selection*. Master's thesis, Iowa State University, Ames, IA, 1994.

[16] C. Richter, *Developing bidding strategies for electric utilities in a competitive environment*. Master's thesis, Iowa State University, Ames, IA, 1996.

[17] C. Richter and G. Sheblé, "Genetic Algorithm Evolution of Utility Bidding Strategies for the Competitive Marketplace." 1997 IEEE/PES Summer Meeting, PE-752-PWRS-1-05-1997. New York: IEEE

[18] G. Sheblé, "Priced Based Operation in an Auction Market Structure". Paper presented at the 1996 IEEE/PES Winter Meeting. Baltimore, MD, 1996.

[19] G. Sheblé, "Electric energy in a fully evolved marketplace." Presented at the 1994 North American Power Symposium, Kansas State University, KS, 1994.

[20] G. Sheblé, and J. McCalley, "Discrete auction systems for power system management." Presented at the 1994 National Science Foundation Workshop, Pullman, WA, 1994.

[21] G. Sheblé "Simulation of discrete auction systems for power system risk management." Frontiers of Power, Oklahoma, 1994.

[22] G. Sheblé, M. Ilic, B. Wollenberg, and F. Wu. Lecture notes from: *Engineering Strategies for Open Access Transmission Systems,* A Two-Day Short Course presented Dec. 5 and 6, 1996 in San Francisco, CA.

[23] G. Thompson, and S. Thore, *Computational Economics: Economic Modeling with Optimization*. San Francisco, CA: Scientific Press, 1992.

[24] L. Tesfatsion, "A Trade Network Game with Endogenous Partner Selection." *Economic Report Series*, Department of Economics, Iowa State University, Ames, IA, 1995.

[25] A.Vojdani, C. Imparto, N. Saini, B. Wollenberg, and H. Happ, "Transmission Access Issues." Presented at the 1995 IEEE/PES Winter Meeting, 95 WM 121-4 PWRS. New York: IEEE, 1994.

[26] A. Wood, and B. Wollenberg, *Power Generation, Operation, and Control*. New York, NY: John Wiley & Sons, 1984.

Charles W. Richter, Jr. received his BS-EE from South Dakota State University in 1992, and his MS-EE from ISU in 1996. He is presently working towards a Ph.D. degree at Iowa State University. Interests include economics of power system operations, economics, and complex adaptive systems.

Gerald B. Sheblé is a Professor of EE at ISU, Ames, IA. Dr. Sheblé received his BS and MS degrees in EE from Purdue University, and his Ph.D. in EE from Virginia Polytechnic. His more than fifteen years of industrial experience include projects with public utilities, research and development, computer vendors and consulting firms. His research interests include power system optimization, power system economics, scheduling and control.

Daniel Ashlock is an Associate Professor of Mathematics at ISU. He earned his BS degrees in math and in computer science from the University of Kansas in 1984. In 1990, Dr. Ashlock received his Ph.D. in math from CalTech. His research and academic interests include combinatorics, graph theory, computer applications to discrete mathematics, and complex adaptive systems.