A Model of Stock Market Participants

Michael de la Maza and Deniz Yuret

Numinous Noetics Group
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
mdlm@ai.mit.edu, deniz@ai.mit.edu

Abstract

In this chapter we describe a stock market simulation in which stock market participants use genetic algorithms to gradually improve their trading strategies over time. A variety of experiments show that, under certain conditions, some market participants can make consistent profits over an extended period of time, a finding that might explain the success of some real-world money managers.

These experiments suggest a four parameter model of market participants. Each participant can be described along four dimensions: information set, constraint set, algorithm set, and model set. The information set captures what data the participant has access to (e.g., the participant has access to all historical price data). The constraint set describes under what restrictions the participant operates (e.g., the participant can borrow money at 1% above the prime rate). The algorithm set indicates what programs the participant can use (e.g., the participant is restricted to hill-climbing optimization algorithms). The model set specifies the language which the participant employs to describe its findings (e.g., the participant uses stochastic differential equations). This four parameter model explains the relative strengths and weaknesses of market participants. After describing the market participant model, we briefly turn to a critique of neural networks, which are the most widely used artificial intelligence tools for financial time series analysis.

We have applied some of the insights that we have gained from doing this and related research to our own trading accounts. We participated in the 1993 U.S. Investing Championships (options division) and finished with a 43.9% return over a period of four months. To leverage this success we have formed a money management firm, called Redfire Capital Management Group, that employs evolutionary algorithms to create fully automated trading strategies for bond, currency, and equity markets. Redfire Capital Management Group launched the first hedge fund that exclusively employs genetic algorithms to create computerized trading strategies on April 3, 1995.

Keywords: Simulation, genetic algorithm, futures markets

1 Motivation

At the turn of the century, Texas wildcatters would decide were to dig for oil by taking a pinch of sand from the ground and tasting it. Their methods were uninformed by any understanding of the causal processes that led to the formation of oil deposits. Everett Lee De Golyer, known as the Father of American Geophysics, was the first to introduce scientific methods into the oil discovery field on a wide scale. As a result, he became a multimillionaire in an age in which a million dollars was still a lot of money.

The money management industry today is similar to the oil industry at the turn of the century. Virtually all of the trillions of dollars invested in today's financial markets are managed by seat of the pants methods that are similar to digging for oil by tasting sand. However, in recent years a growing number of individuals and companies have dedicated themselves to applying the methods of science and engineering to the stock market. Those who have been successful in doing so have made fortunes and there are many more fortunes to be made.

In this chapter, we take another step in this direction by applying our understanding of genetic algorithms to the market. This chapter describes a simulation of a market in which individuals evolve trading strategies and compete against each other to maximize profits.

2 Justification

Why simulate instead of studying real markets? Simulations have several advantages over direct observation:

- The parameters of the simulation can be endlessly modified in order to provide greater understanding.
- Simulations are typically much faster than real markets, therefore many more experiments can be done.
- The investigator has very fine-grained control over the simulation.

However, there are some disadvantages associated with simulations:

- Care must be taken when carrying over the results of a simulation to real markets. They may not apply because all models make assumptions about real markets which may be incorrect.
- A simulation necessarily involves abstraction, and this abstraction may leave out some of the most important elements of the market. Of course, this would not occur intentionally but given the complex and little understood structure of the markets it may happen unintentionally.

- A trader who makes a million dollars in a simulation has not made a million dollars!

The simulation described here does not capture all of the subtleties and nuances of the market. Its implementation contains many simplifications and assumptions. Nevertheless, it gives insight into how some market participants may be able to consistently uncover profit-making opportunities.

This chapter begins by discussing the details of this simulation and three experiments with the simulation are described and analyzed. Based on the data from these experiments, we suggest a four parameter model of participants that we have found helpful in characterizing what are the relative advantages and disadvantages of a market participant. The chapter concludes with a brief criticism of the standard application of neural networks to financial time series analysis.

3 Simulation details

Our market simulation consists of a series of days. A day contains four steps:

1. Each participant computes a fair price for a security. In our simulation, there is only one security in the entire market and each participant is forced to submit a bid during each day.
2. An equilibrium price, which balances the buyers and sellers, is computed from the fair prices submitted by each participant. The equilibrium price is the median of the fair prices.
3. Participants whose fair prices are above the equilibrium price buy at the equilibrium price, and participants whose fair prices are below the equilibrium price sell at the equilibrium price. Thus, for every security that is sold, a security is bought and so one participant's gain is another participant's loss. No transaction fees are charged. All traders buy or sell a single contract. They may not vary the trade size.
4. Participants improve their bidding strategies by using a genetic algorithm.

Each individual has a *visible strategy* which it uses to compute the fair price. In addition, each individual has a set of *invisible strategies* that cannot be directly perceived by other market participants. The visible strategy is the strategy that performed the best during the last day and it is updated every day.

The strategies that individuals can learn are very simple. They consist of a quadruple $< a, b, c, d >$ of four real numbers and the fair price is computed by the formula:

$$f(a, b, c, d) = a * x + b * y + c * z + d$$

where x is the ten day moving average of the equilibrium prices; y is the last equilibrium price; and z is 1 if the last change in equilibrium prices is positive, -1 if it is negative, and 0 otherwise.

Suppose, for example, that the visible strategy of a market participant is $< 1, 2, 3, 4 >$ and that the ten day moving average is 12, the last equilibrium price is 11, and the last change in equilibrium prices is negative. Then, the fair price compute by this strategy is: $1 * 12 + 2 * 11 + 3 * -1 + 4 = 35$.

If there are four participants which submit fair prices of 33, 34, 35, and 36 then the equilibrium price is 34.5. The two participants who submitted fair prices below 34.5 sell the security at 34.5 and the two participants that submitted fair prices above 34.5 buy the security at 34.5.

The participants are forced to liquidate their holdings at the equilibrium price of the next round. So, if the equilibrium price is above 34.5, then the two participants who purchased the security at 34.5 will make money and the two participants who sold the security will lose money. On the other hand, if the equilibrium price is below 34.5 the two participants who sold the security at 34.5 will make money and the two participants who purchased the security will lose money.

The strategies maintained by each participant are modified and improved by a genetic algorithm [Hol75, Gol89] with traditional mutation and crossover operators. The mutation operator adds a number uniformly distributed between -.1 and .1 to a coefficient and the crossover operator is traditional single point crossover. The best strategies are selected and copied using standard proportional selection with a constant offset that ensures that all fitness values are positive [Mic92]. The fitness values are calculated by subtracting the fair value computed by a strategy from the spot price on the next day.

4 Experiments

This section describes three experiments that we have done to explore various facets of our simulation. In all of the experiments, there are four participants who trade.

4.1 Effect of noise

Some market pundits say that the market makes many unexplainable moves that will eventually wipe out all strategies. Separating the signal from this noise is thought to be one of the hardest problems facing all traders.

Can market participants still uncover good market strategies in the face of noise? That is the question that this experiment is designed to test. In

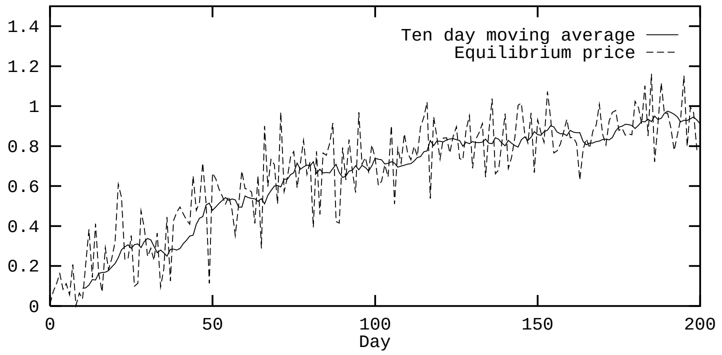

Figure 1: Equilibrium price as a function of time. The ten day moving average, the last equilibrium price, and the sign of the last change are all random numbers between -5 and 5.

this experiment, the ten day moving average, the last equilibrium price, and the sign of the last change are all replaced with random numbers between -5 and 5 and the equilibrium price is set to 1. Thus, the optimal strategy is <0, 0, 0, 1>, which reflects the decision to ignore the three noisy variables. We chose to add the noise directly to the input variables instead of to the output to make the simulation conceptually clearer.

A ten day moving average of the equilibrium prices is shown in Figure 1. As you can see, the equilibrium price slowly but surely moves towards 1, indicating that the participants are able, over time, to ignore the noise in the information provided to them.

4.2 Forced liquidations

This experiment tests what effect forced liquidations have on profits. Market participants must sometimes liquidate their holdings for a variety of reasons – they get hit with lawsuits, they want to send their children to college, or they have substantial losses in real estate.

In this experiment, a participant, with probability .2, is forced to report a fair price that is one half of the fair price computed by its visible strategy. This gives market participants who do not have to liquidate their holdings the opportunity to make substantial profits. Figure 2 shows how the profit of the most profitable participant changes over time. Initially, the profit rises and then drops sharply, but it never reaches zero as the participants who do not have to liquidate take advantage of those who do.

In contrast, Figure 3 shows the profit in an experiment in which participants are not forced to liquidate. In this case, the profit quickly approaches zero as the participants lock in on optimal trading strategies.

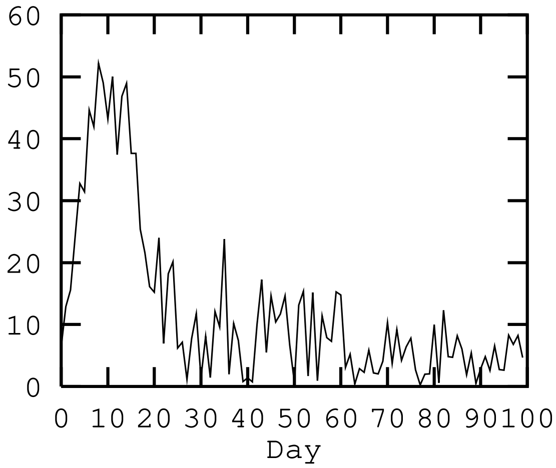

Figure 2: Profit of most profitable individual as a function of time. This data is averaged over ten runs of the simulation. Because participants are forced to liquidate their holdings with some probability, the profit does not approach 0.

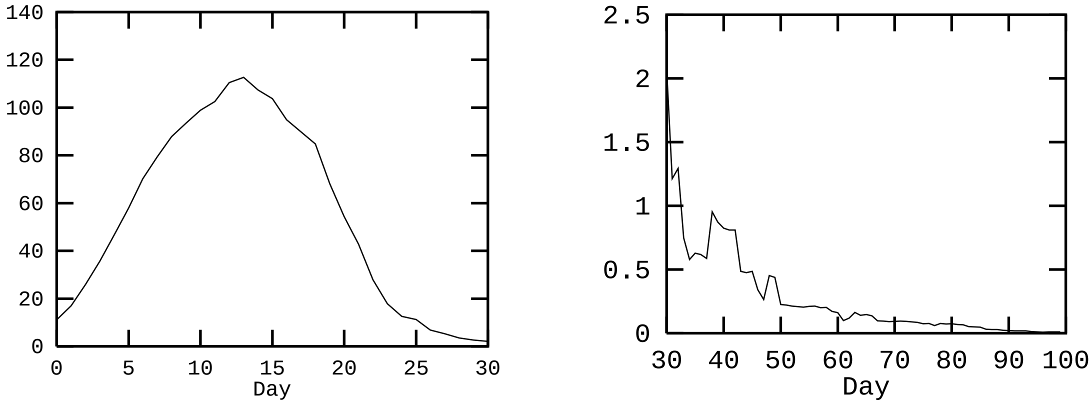

Figure 3: Profit of most profitable individual as a function of time. This data is averaged over ten runs of the simulation. The profit approaches 0 because the participants are able to learn optimal strategies.

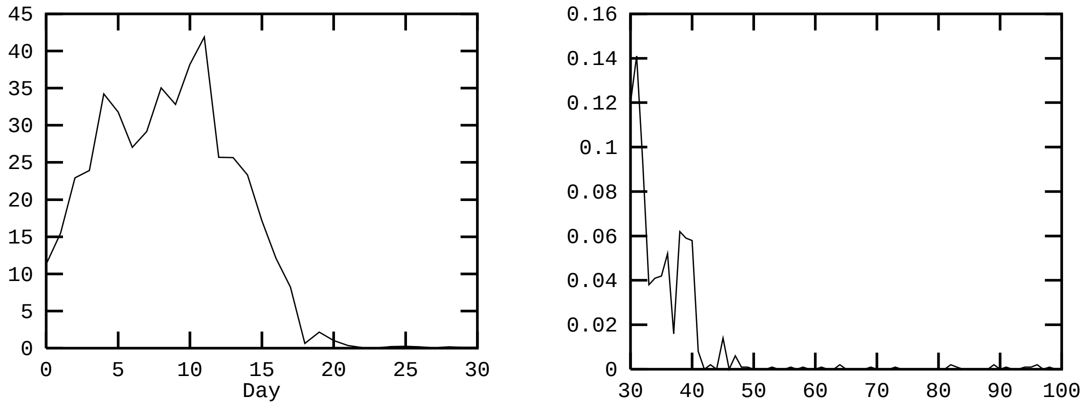

Figure 4: Profit of most profitable individual as a function of time. This data is averaged over ten runs of the simulation. Profit quickly approaches 0 because the participants are sharing their visible strategies.

4.3 Sharing strategies

In our final experiment we studied what happens to profitability when the market participants share their visible strategies. After each day each participant's visible strategy is added to the list of strategies of all of the other participants. As shown in Figure 4 the profitability of the most profitable experiment drops sharply. This figure should be compared to Figure 3 which shows the profitability of the most profitable participant when no sharing occurs.

This experiment helps to explain why most of the results published in the field of stock market analysis are negative. Sharing profitable strategies really does reduce profitability for everyone.

5 Understanding market participants

As a result of these experiments we have arrived at a four parameter model of market participants. Each market participant can be characterized along these four dimensions:

- Constraint set. Is the participant beholden to shareholders or investors? Do profits have to be reported on a monthly, quarterly, or yearly basis? At what rate can the participant borrow capital? What securities and markets can the participant trade?

- Information set. Does the participant have access to end of day data, minute data, or tic data? Is information about order flow available? Does the participant have access to floor traders or market makers?

- Algorithm set. Does the participant use pencil and paper to explore the space of trading strategies? Backpropagation? Genetic algorithms? Dynamic hill climbing?
- Model set. Is the participant restricted to linear models? Can if-then rules, previous market scenarios, and stochastic differential equations all be employed to represent market knowledge?

At our own money management firm, Redfire Capital Management Group, we only trade markets in which we have a considerable advantage along two or more of these four dimensions.

6 Two shortcomings of neural networks

Neural networks are the most widely used artificial intelligence method for financial time series analysis and thus they are the main competitors to the form of research explored in the previous sections of this chapter. In this section we describe two shortcomings of the standard neural network application. First, backpropagation search takes place in sum of squared errors space instead of risk-adjusted return space. Second, the standard neural network has difficulty ignoring noise and focusing in on discoverable regularities.

6.1 Sum of squared errors vs. risk-adjusted return

> My financial success stands in stark contrast with my ability to forecast events.
>
> - George Soros[Sor94, page 301]

We present two trading strategies which have the property that one is superior when sum of squared errors is the utility measure but is inferior when risk-adjusted return, as defined by the Sharpe ratio, is the utility measure. Because what matters to investment managers and their clients is risk-adjusted return, this suggests that the standard neural network application, which minimizes the sum of squared errors, should be retooled.

Consider the time series: 10,20,30,60,10,20,30,10. This can be thought of as the price of a security on consecutive days. If the neural network is asked to predict the next element in this series given the three previous elements then there are a total of four unique input sequences and five input/output pairs, as shown in Table 1. This table shows two strategies, strategy A and strategy B, which produce the same output for three of the four input sequences, but differ on the only sequence which is not followed by a unique element (the first and last rows in Table 1). Strategy A is the optimal sum of squared errors strategy but strategy B has a better Sharpe ratio, as shown in Table 2. This comparison assumes that the strategy buys when the predicted

Sequence (input)	Next element (output)	Strategy A Prediction	Strategy A Profit	Strategy B Prediction	Strategy B Profit
10,20,30	60	35	30	30	0
20,30,60	10	10	50	10	50
30,60,10	20	20	10	20	10
60,10,20	30	30	10	30	10
10,20,30	10	35	-20	30	0

Table 1: The five sequences in the time series 10,20,30,60,10,20,30,10. The neural network is asked to predict the next element in the sequence given the previous three elements. Strategy A is the prediction made by the strategy which minimizes the sum of squared errors. Strategy B is a strategy which has a higher sum of squared errors, but a lower Sharpe ratio, than strategy A, as shown in Table 2.

	sum of squared errors	μ	σ	Sharpe ratio ($\frac{\mu}{\sigma}$)
Strategy A	1250	16	23.324	0.686
Strategy B	1300	14	18.547	0.755

Table 2: Comparison of strategy A and strategy B. Strategy A has a lower sum of squared errors, but strategy B has a better Sharpe ratio (the risk-free rate is assumed to be 0%).

value of the next element is greater than the last element of the sequence, sells when the predicted value of the next element is less than the last element of the sequence, and does not trade when these two values are the same. A constant amount is traded each time and the market is assumed to be frictionless.

A skeptic would argue that the flaw in this example is that a *forecasting* strategy should not be confused with a *trading* strategy. Such a skeptic would say that once the neural network has been trained to be a good *forecasting* strategy then some sort of post-processor should be added to the output to turn it into a good *trading* strategy. This appears to be a somewhat circuitous approach: Why search in the wrong space (sum of squared errors) and then repair, when the search can be done in the space of interest (risk-adjusted return)?

6.2 The importance of knowing what you don't know

The standard application of neural networks forces a prediction to be made for every input sequence. Because of this, the neural network must distribute its representational capacity across the entire time series, instead of being able to focus in on regions which have discoverable regularity. As a result, the neural network is sometimes incapable of uncovering simple regularities because its representational capacity is inappropriately employed.

This point is illustrated by the function shown in Figure 5(A) which has three segments: the first (points 0 to 49) and third (points 100 to 149) segments were generated by randomly choosing numbers between 0.25 and 0.75 and the second segment (points 50 to 99) is the line $y = x/100 - .25$. A wide variety of neural networks with varying numbers of hidden units (0, 2, 5, 10, 20, and 50), different learning rates (.01, .001, and .0001), and different inertia parameters (0, .1, .2, .3, .4, .5, .6, .7, .8, and .9) were trained to predict the value of point n given points $n-5$ through $n-1$. The predictions made by one representative neural network are shown in Figure 5(B). The predictions made by the same neural network when trained only on the second segment of the function shown in Figure 5(A) are shown in Figure 5(C).

A comparison of these two graphs shows that the prediction of the middle segment is much more accurate in Figure 5(C) than in Figure 5(B). Why? Because part of the neural network's representational capacity has been spent trying to model the first and third segments, even though these segments do not have discoverable regularity. Importantly, the function in Figure 5(A) is extremely simple. The regularities in real financial time series are significantly more subtle and the ability to distinguish between the knowable and unknowable becomes significantly more important.

The underlying problem on which this example stands is the requirement that the standard neural network make a prediction at every data point: "don't know" is not an option. However, not only does the capability to

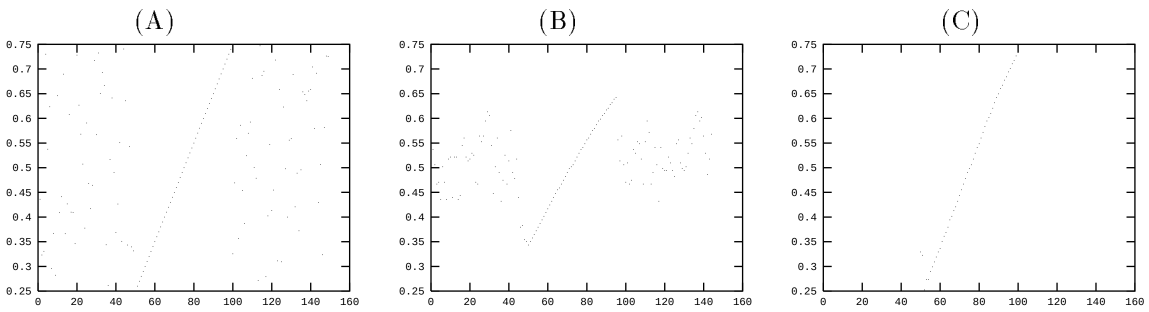

Figure 5: (A) Target function. The neural network is asked to predict the value of point n and is given as input the value of points $n-5$ through $n-1$. The first 50 points are random numbers between 0.25 and 0.75 as are the last 50 points. The middle 50 points satisfy the linear equation $y = x/100 - .25$. (B) Output of a neural network with five hidden units trained on the target function. The inertia parameter was set to 0.9 and the learning rate was set to .001. The neural network was trained for one million epochs. This output graph is representative of the performance of many neural networks that we tested on this function. (C) Output of a neural network with five hidden units trained on the middle 50 points of the target function. The inertia parameter was set to 0.9 and the learning rate was set to .001. The network was trained for one million epochs. Compare to the middle 50 points in (B).

say "don't know" free up representational capacity, but, as in the case of Figure 5(A) over the interval [0,49] and [100,149], it is sometimes the best possible model of the data given the representational capacity of the neural network. Many statistical methods, such as linear and multiple regression, share this shortcoming with the standard neural network application. As a result, these methods are, in effect, unable to say what human traders often do: "Current market conditions are beyond my comprehension, therefore I am not going to try to understand them." For the graph shown in Figure 5(A) this is *the* appropriate response for the first and third segments.

6.3 Possible solutions

Our view is that the shortcomings discussed above are potential showstoppers which demand the full attention of those who are interested in applications of neural networks to financial time series analysis. In this section we speculate on ways of overcoming these limitations.

One way to address both problems is to change the search algorithm from backpropagation to simulated annealing [KGV83, Ing93], genetic algorithms [Hol75, Gol89], standard gradient descent, or any one of the many optimization methods which allow search over arbitrary utility functions. Adopting this solution would allow the direct implementation of risk-adjusted return as the utility function and the neural network would no longer be forced to make a prediction for every input. Under this scheme, the representation (or model) would still be sums of nested sigmoidal functions.

Finessing the data so that minimizing the sum of squared errors is equivalent to maximizing risk-adjusted return is another way to address the first shortcoming. Success in finding such a transformation would probably be helpful in finding solutions to other problems not associated with financial time series analysis.

A possible solution to the second problem has been explored by Jordan and Jacobs [JJ93]. They suggest creating a hierarchical mixture of neural networks so that each neural network can become an expert at identifying the regularities in one component of the data.

7 Related work

Nottola, Leroy, and Davalo [NLD92] describe an artificial market system in which agents (called participants in this paper) learn rules that predict movements in price. Much of their paper describes different approaches to modeling markets, but they do discuss one experiment which compares the performance of agents with different learning strategies and they conclude that adaptive agents out-perform non-adapting ones.

LeBaron [LeB94] has developed a stock market simulation in which participants learn strategies using a genetic algorithm, as they do in this paper.

The strategies of his market participants are encoded as bitstrings in which each bit corresponds to a boolean variable that is precomputed. His system features a mechanism for easily changing the risk aversion of individual participants. LeBaron notes [Fre93] that even if there are regularities in real market data that can be exploited, the data might contain so much noise that the regularities are barely perceptible.

There is substantial evidence that the currency futures market is not efficient [Tho86, Cav87, Gla87, HKY90]. As empirical evidents mounts against the view that markets are efficient, theoreticians will be forced to provide alternative explanations of the market's behavior. One of the most likely candidates for this is Simon's bounded rationality theory [Sim57, Sim82]. Sargent [Sar93, page 4] explores markets in which the participants have bounded rationality and writes that "the bounded rationality program wants to make the agents in our models more like the econometricians who estimate and use them."

We have not compared our genetic algorithm approach to neural network approaches because neural network training time is prohibitive.

8 Conclusion

We plan to extend the simulation described in this chapter in at least two ways. First, the algorithm and model sets of all of the participants in these experiments are the same – only the constraint sets and information sets change. For example, in one experiment the constraint set of participants includes forced liquidations while in other two experiments it does not. Likewise, in one experiment the participants have access to uncorrupted moving average information while in the other two they do not.

In the future we will pit participants with different algorithm sets and model sets against each other. Second, this simulation contains only one security and all of the participants are required to purchase or sell a security during every round. Our future market simulations will have multiple securities and the participants will have the option of not trading.

We have applied some of the insights that we have gained from doing this and related research to our own trading accounts. We participated in the 1993 U.S. Investing Championships (options division) and finished fifth with a 43.9% return over a period of four months. To leverage this success we have formed a money management firm, called Redfire Capital Management Group, that employs artificial intelligence methods, including simulations similar to the ones described here, to create fully automated trading strategies for interest rate, currency, and equity markets. We trust that other market aficionados will find these simulations equally helpful.

9 Acknowledgments

Thanks to Steve Baron, Tom Berghage, Prof. Biethahn, Jim Hutchinson, Michael Jones, and Dr. Volker Nissen. Some of the neural network simulations were performed on the Cray C90 at the Pittsburgh Supercomputing Center. This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-92-J-4097 and by the National Science Foundation under grant number MIP-9001651.

References

[Cav87] K. Cavanaugh. Price dynamics in foreign currency futures markets. *Journal of International Money and Finance*, 6(3):295–314, 1987.

[Fre93] D. Freedman. Enter the market merlins. *Forbes ASAP*, 1993.

[Gla87] D. Glassman. The efficiency of foreign exchange futures markets in turbulent and non-turbulent periods. *The Journal of Futures Markets*, 7(3):245–267, 1987.

[Gol89] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[HKY90] G. Harpaz, S. Krull, and J. Yagil. The efficiency of the U.S. dollar index futures market. *The Journal of Futures Markets*, 10(5):469–479, 1990.

[Hol75] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[Ing93] L. Ingber. Simulated annealing: Practice versus theory. *Mathematical and Computer Modeling*, 18(11):29–57, 1993.

[JJ93] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. Technical Report 1440, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[KGV83] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[LeB94] B. LeBaron. An artificial stock market. MIT AI vision seminar colloquium, 1994.

[Mic92] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 1992.

[NLD92] C. Nottola, F. Leroy, and F. Davalo. Dynamics of artificial markets: Speculative markets and emerging 'common sense' knowledge. In *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 185–194, 1992.

[Sar93] T. J. Sargent. *Bounded Rationality in Macroeconomics*. Clarendon Press, Oxford, 1993.

[Sim57] H. Simon. *Models of Man: Social and Rational; Mathematical Essays on Rational Human Behavior in Society Setting.* John Wiley, New York, 1957.

[Sim82] H. Simon. *Models of Bounded Rationality.* MIT Press, Cambridge, MA, 1982.

[Sor94] G. Soros. *The Alchemy of Finance: Reading the Mind of the Market.* John Wiley & Sons, Inc., New York, 1994.

[Tho86] L. Thomas. Random walk profits in currency futures trading. *The Journal of Futures Markets*, 6(1):109–125, 1986.