Interactive Inversion of Financial Markets Agent-Based Models

Trent Ashburn, Eric Bonabeau
Icosystem Corporation
10 Fawcett Street
Cambridge, MA 02138
Email: {trent, eric}@icosystem.com

***Abstract*-Two financial markets examples illustrate how Interactive Evolutionary Computation (IEC) can be used for agent-based model inversion. The first example shows that IEC can be used to discover a model that reproduces synthetic data generated with a very simple model. The second example applies IEC to the inversion of a more sophisticated agent-based model to reproduce real data from a specific market event.**

I. Introduction

There have been a number of attempts over the last decade to model financial markets with agent-based modeling (ABM) [1-8]. In ABM, systems are modeled as collections of autonomous decision-making entities, called agents. Each agent individually assesses its situation and makes decisions based upon a set of rules. Agents may execute various behaviors appropriate for the system they represent –for example, buying or selling. Repetitive interactions between agents are a feature of ABM, which relies on the power of computers to explore dynamics out of the reach of pure mathematical methods [9]. The power of ABM lies in their ability (1) to let the modeler describe behavior in very natural terms and (2) to capture emergent phenomena. A financial market seems to be a natural fit for ABM. The dynamics of the stock market results from the behavior of many interacting agents, leading to emergent phenomena that can be understood using a bottom-up, ABM approach.

While ABM is useful in producing market-like aggregate-level patterns from individual-level rules, the main issue in financial markets ABM is calibration and validation: how can one evaluate the quality of a model, from both a structural perspective (how sound is the model?) and from an econometric perspective (how well does that model reproduce the data quantitatively?). In practice, calibration and validation of financial markets ABM are often neglected. One reason is that the use of purely numerical scoring methods to evaluate data fit and guide the search for explanatory models constrains the search path so dramatically that no good fitting model is found or the best fit is generated by a low-plausibility model. In many situations the fitness function for a model cannot be practically formulated mathematically. This problem can be overcome if one can allow more subjective factors to guide the search for "good" models, enabling ABM users to integrate financial economics expertise into their models.

Boschetti & Moresi [10,11] have proposed to replace the numerical evaluation of data fit by a subjective evaluation. A technique originally developed to generate "interesting" images and pieces of art [12-16] is used to perform model inversion by integrating subjective knowledge into the evaluation process. The technique (see [17] for a review) is a directed search evolutionary algorithm which requires human input to evaluate the fitness of a pattern (here, the fitness might be how well the model reproduces the data qualitatively) and uses common evolutionary operators such as mutation and crossover to breed the individual-level rules that produced the fittest collective-level patterns. Interactive evolutionary computation (IEC), as this technique is known, combines computational search with human evaluation [17].

In this paper we present an application of IEC to financial markets ABM inversion. Two examples illustrate how IEC can be used to discover the parameters of an agent-based model of a financial market from aggregate observations. In both experiments, the user operates a visualization tool to navigate a parameter space using selection and variation operators. Parameter values define traders and their trading strategies which, in turn, generate a synthetic price history. The user's goal is to find a combination of parameter values that can reproduce a target price history.

The first experiment uses a simple model to generate a synthetic target price history. Because the model is non-stochastic and the search space is small, a perfect visual match to the model's output can be found.

The second, more complex experiment utilizes a target price history from a market frenzy that occurred on the London Stock Exchange in September 2002. IEC is used in this case to obtain a qualitative fit.

II. Agent-Based Model

A simple model of a financial market is used in both experiments. Each model simulation includes its own order management and clearing mechanism (an order book), traders operating trading strategies, a market maker posting orders on the book which are matched with traders' orders, and a price history. At the end of each time step, after all trading and market-making, the arithmetic mean of the best bid and best ask is appended to a simulation's price history.

A. The order book

Visualized in Figure 1, the order book matches orders in a continuous double-sided auction. It holds and clears both limit orders and market orders, where clearing is the action of matching a bid with a same-priced ask and removing these from the book. Same-priced limit orders are matched with other limit orders or market orders. In the process of clearing, the size (number of shares) of each matched order is reduced by the amount of the smaller-size order. Orders with size zero are removed from the book. Upon order submission, the book immediately clears any and all clearable orders. Orders are characterized as follows: (i) Bid: an offer to buy; (ii) Ask: an offer to sell; (iii) Best bid: the

0-7803-8515-2/04/$20.00 ©2004 IEEE

highest bid; (iv) Best ask: the lowest ask; (v) Limit order: a bid or ask at a specific price or better; (vi) Market order: an offer to buy at the best ask or sell at the best bid.



IBM

GET STOCK IBM go

Symbol Search

LAST MATCH		TODAY'S ACTIVITY	
Price	92.3800	Orders	5,416
Time	12:52:43.307	Volume	31,764

BUY ORDERS		SELL ORDERS	
SHARES	PRICE	SHARES	PRICE
100	92.3000	300	92.3700
300	92.2200	300	92.4300
600	92.1100	700	92.6900
200	92.0700	821	92.8400
2	91.5000	1,000	93.1900
700	91.4300	700	93.3000
1,000	91.3700		
30	90.7700		
30	87.7900		
3	71.8800		

Figure 1. A real-world continuous double auction limit order book from the ECN, Island (http://www.island.com).

B. Traders

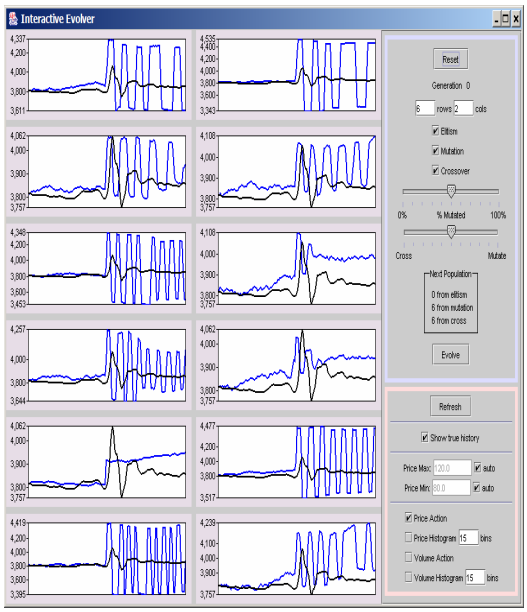
In each time step, traders take turns trading according to their strategies. Traders trade once per time step, each trader submits only market orders, and traders' strategies (described in each experiment) are based upon a set of initial conditions and the accumulated price history of the stock. In each time step, after each trader trade, the market maker trades.

C. Market maker

To ensure liquidity in the market, there is one market maker in each simulation, tightening the spread and maintaining order depth. The *spread* is the difference between the best bid and the best ask. When a trader trades, bids or asks are often cleared from the book, which widens the spread in a particular direction. *Spread tightening* is the action of the market maker adding bids if the trader had bought, or adding asks if the trader had sold, until the spread is a specific dollar amount. In our experiments, the market maker separates all its orders by $1 increments, and attempts to maintain a spread of $1. *Order depth* refers to the number of uniquely-priced bids and asks that the market maintains on the book. This buffers trading and must be of a sufficient depth to handle all trading; several hundred orders suffice in our experiments.

III. Interactive Evolution

The IEC search method employs a genetic algorithm and a graphical user interface to facilitate the user acting as the fitness function. A small initial population of agent based models is generated with random parameter values. The resulting price histories are generated by running the models, and then shown to the human observer. The observer selects interesting patterns according to whatever objective and subjective criteria the observer may be using to visually compare candidates with the target. These selections are considered the fittest individuals of the generation. The user configures a set of operators - elitism, mutation, and crossover – which are used to produce a new generation of models from the user-selected "fittest" individuals in the previous generation. The new generation is then simulated and the resulting price histories are again presented to the observer. This procedure is iterated until interesting patterns emerge from the search that more closely match the target. Over multiple generations the population may converge toward the target.



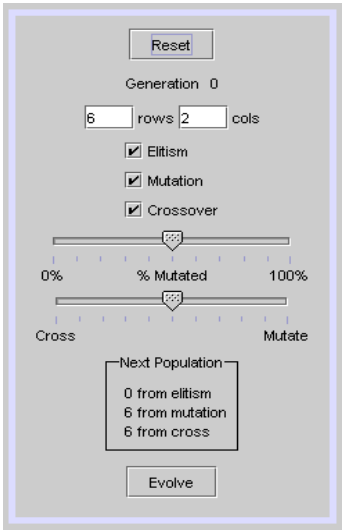


Figure 2. User interface and user control panel.

A. User interface

The user interface, shown in Figure 2, is a critical component of the method, which depends crucially upon the user's ability to evaluate visualizations of the candidate solutions [17] –obviously this method can only work if the population size is kept small and if interesting patterns emerge after a reasonably small number of generations. The user interface shows the price histories of each candidate model in the left window, each overlaid on top of the target price history. On the right are two controls panels: (1) the main IEC control panel configures the size of the left-hand visualization grid, toggles operators, controls the chance that a gene is mutated, and controls the proportion of crossover versus mutation; the evolve button produces the next generation; (2) the second control panel controls model drawing.

B. Evolutionary algorithm

Each model simulation and its price history can be considered a phenotype, generated from the set of trading strategies and their parameters used in the simulation (the genotype). The parameters of these strategies vary across genotype, and it is the composition of these genotypes that we are interactively evolving. The genotype of a model has genes that are numbers – initially random numbers. Each number codes for a specific aspect of the set of traders (how many of each kind of trader, parameters of each strategy) and has bounds specific to the aspect it codes for. For example, a gene that codes for the number of traders might be bounded between 10 and 100. Candidates are either selected or not, and non-selected candidates are discarded. The user may turn on and off each of three operators to produce next generation members - elitism, mutation, and crossover. Elitism is applied first; then crossover and mutation are performed on randomly chosen selected members to produce remaining members. The number of new members produced by crossover versus the number produced by mutation is a proportion chosen by the user.

Elitism. Selected candidates are copied to the new generation.

Mutation. The user controls the chance that each gene is mutated - between 0% and 100%. As the mutation algorithm iterates through each gene, a random number between 0.0 and 1.0 is chosen. If that number is less than the mutation percentage (expressed as a decimal, 50% is 0.50), that gene is mutated to a new value within the bounds of that variable. The specific algorithms used for mutation are described with each experiment.

Crossover. Double-point crossover chooses crossover points in a uniform random fashion and produces a new candidate by recombining two parent genotypes from among the selected candidates. The specific algorithms used for crossover are described with each experiment.

IV. Experiment 1

In this model, during each time step, a trader can buy, sell, or do nothing. Each trader can only sell as much as the trader has bought. Each genotype, including the target genotype, contains six initially-random real-number values that code for price thresholds for three traders, which trigger buying and selling. Each trader has two thresholds – an upper and a lower, bounded between $90 and $110. Each trader's trading strategy works as follows: (i) if $p<=L$, switch to buy mode: buy one stock each time step at market price; (ii) if $p>=U$, switch to sell mode: sell one stock each time step at market price. p is the most recent price in the price history, L is the trader's lower threshold, and U is the trader's upper threshold. The market marker's initial best bid is $100 the best ask is $101, and all the market maker's orders are limit orders of size 1. Each simulation is run for 20 time steps and the user reviews 12 candidates in each generation.

A. Operators

Mutation Operators. When visually inspecting a graph, a user might select a genotype because it had a threshold close to a target threshold. When mutating, however, this evolutionary guidance is poorly reflected in how that gene is modified. If mutated, the new value of the gene for this threshold, v_n jumps to a bounded uniformly random value, $v_n=min+rand(max-min)$, where min is the gene's lower bound, max is the upper bound, and rand is a random number in the interval [0,1]. Note that the value does not jump with higher probability to a value nearby the pre-mutated value. To increasing the likelihood that a mutation fulfills our user's intention, the probability distribution of the new value was changed from uniform-random to triangle-shaped with vertices on the lower threshold bound (90), on the upper threshold bound (110), and on the pre-mutated value. The calculation for the triangle-shaped distribution: if $rand_1<0.5$ then $v_n=v_{old}+|rand_2-rand_3|(max-v_{old})$, else $v_n=v_{old}+|rand_4-rand_5|(v_{old}-min)$, where v_n is the new value, v_{old} is the old value, min is the lower bound, max is the upper bound, and $rand_1$ thru $rand_5$ are random numbers in the interval [0,1].

Crossover Operators. The initial crossover operator chose parent genotypes in uniform random fashion from among those selected by the user, then a single crossover spot was selected in uniform random fashion and a resulting genotype was produced. In this way, only upper thresholds are crossed with upper thresholds, and only lower thresholds are crossed with lower thresholds.

When visually inspecting a graph, users often desired to effectively "cut and paste" a threshold from one genotype into another – for instance, to produce a new candidate with a lower threshold from one genotype and an upper threshold from another. Unless these thresholds exist at one end of a genotype, they could not be cut and pasted as a module.

Moreover, the order of the genes in each genotype was static and they were crossed in order, which prevented two thresholds that occupied the same location on two genotypes from ever existing in the same genotype– they were mutually exclusive. To the user, this would result in the frustration of being unable to combine two thresholds in two different phenotypes.

Double-point crossover solved the cut and paste problem, and randomly ordering the traders of each parent before crossing solved the mutual exclusivity problem – although this does occasionally result in lower thresholds being higher than upper thresholds, producing a net flat price history in isolation. This case is easily discarded by the user.

Double point crossover proceeds as follows: one of the selected parents is randomly chosen as parent1, another as parent2. The traders of each parent are ordered randomly and their thresholds are assembled in upper, lower order into genotypes g1 and g2, respectively: $g_1=\{t1_U,t1_L,t2_U,t2_L,t3_U,t3_L\}$ and $g_2=\{t4_U,t4_L,t5_U,t5_L,t6_U,t6_L\}$, where *t1* thru *t6* are traders, and *U* and *L* refer to the upper and lower thresholds of each trader. Locations in each set are indexed from 1 to 6.

Two points, *p1* and *p2*, are chosen as random integers in the interval [0,*n*], where *n* is the number of genes – in this case, 6. The new genotype is then composed from g_1 genes in positions $<=p_1$ and $>p_2$, and from g2 genes in positions $>p_1$ and $<=p_2$ (Figure 3).

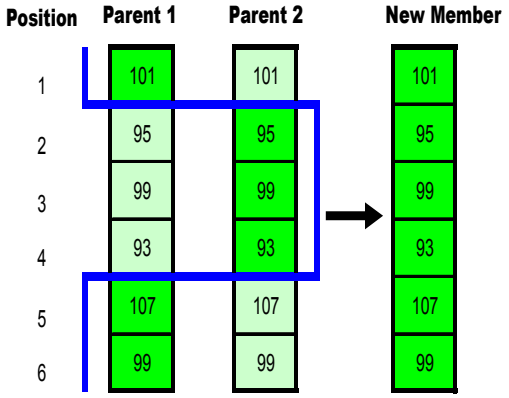


Figure 3 – An example of crossover, with p_1=1 and p_2=4.

While our modifications to the genetic operators made the user's intentions more manifest in the resulting genetic recombination, other user intentions in selecting candidates as "fit" could not be represented in the evolution – for instance, a user may select a candidate for recombination because it is the same shape as the target but spatially displaced, or has the same frequency as the target but has a mis-scaled amplitude. In either of these cases, the user's evolutionary guidance (selecting that candidate) is a local minima; it does not typically bring the population closer to the target because the genetic operators can't perform the user's intention - a spatial shift or vertical scaling. The genetic operators and the user's reasons for selecting a candidate as "fit" should match.

B. Experiment 1 Results

We found it possible, using IEC, to reach an exact visual match with the target over an average of 6 generations in 10 trials from a variety of starting populations. The fewest number of generations to reach the target, on average, was achieved by mutating the population of candidate genotypes until thresholds were discovered that matched thresholds present in the target, even though these thresholds did not necessarily occur in the same candidate genotype. Then, crossover is employed to recombine the proper thresholds until the matching combination is discovered. Figure 4 shows a typical sequence of evolution steps with interactions.

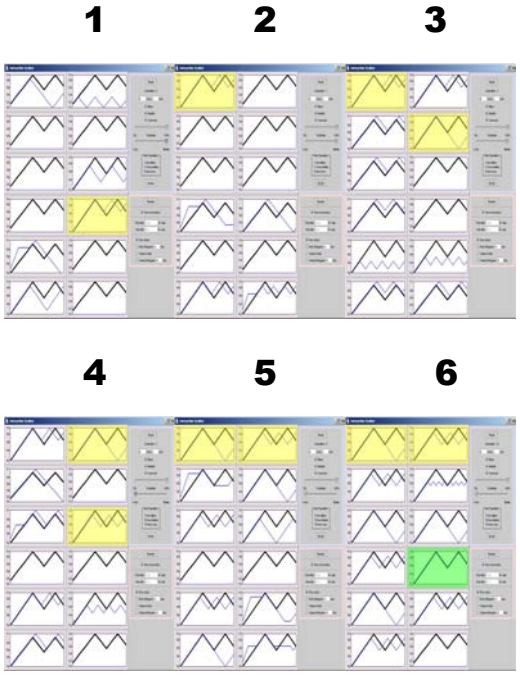


Figure 4. Example sequence of evolutionary interaction steps.

V. EXPERIMENT 2

A market frenzy is a large deviation from the efficient market hypothesis for a relatively short period time, usually triggered by an anomalous event such as a trading error, a glitch in the trade processing information system, a significant piece of news, etc. Although such events are unusual, they do happen. One example is the market frenzy that occurred on September 20, 2002 at the London Stock Exchange and lasted for about 20 minutes and generated losses on the order of £100M for some of the players [18]. The trade volume (£3.2B) of this 20-minute event was larger than the volume of an average trading day. According to Muchnik and Solomon [19], "the trade activity (trades/second) was so high that some computer systems failed to cope and prices of shares were delayed" – probably leading to an amplification effect. The event began at 10:10 am and within 5 minutes the FTSE100 index rose from 3,860 to 4,060. Within another few minutes, the index fell to 3,755, before returning to a value slightly above its original level at the end of the 20 minutes (Figure 5).

Although Muchnik and Solomon [19] tell us that "it is believed that the event was triggered by a huge order (£1.2B) mistakenly submitted twice (or even three times) causing all trade participants to try to exploit the opportunity," the exact course of events is not precisely known; the players don't like to publicize their mistakes. The hypothesis that the event was triggered by a glitch is likely, given the return to a stable state, but according to various private sources the glitch is actually a combination of a trading error and an information system problem. In that context, our formulation of the problem we wish to address is: what is the nature of the initial trigger and what are the

likely behaviors of the different players that can generate such a deviation from the market's stationary behavior? Our model will assume that one large order is being submitted, producing the necessary trigger to push the market out of its stationary state "without affecting its fundamental dynamical parameters", following Muchnik and Solomon's [19] plausible assumption.

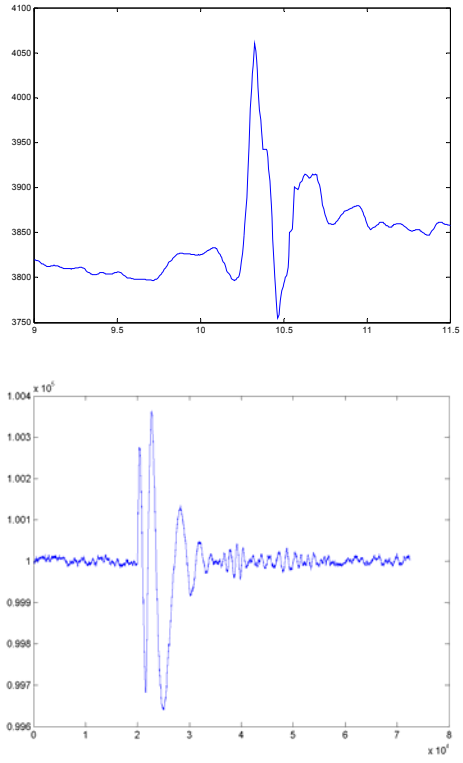


Figure 5. FTSE 100 dynamics following a likely trading mistake on September 20, 2002. From Muchnik and Solomon's model [19] with a trend-dependent panic factor. Modified from [19].

The goal of IEC here is to determine the nature of the herding behavior that could produce the event. While Muchnik and Solomon [19] offer an elegant first-order explanation of the phenomenon (an oscillator model akin to that of a guitar string), their model produces several oscillations around the stationary level before stabilization (Figure 5), while in the real data there is only one oscillation before stabilization. The IEC approach will be used to design a model that produces a better qualitative fit with the real data.

A. Model and genotype representation

Traders in this experiment come in four flavors (Figure 6): fundamentalists, chartists, noise traders, and one whoops trader.

Each has unlimited buying power, and unlimited shorting power (in the real world, shorting is betting that the stock is going down in value, using borrowed stock; covering is buying back the stock and returning it to the lender. For our purposes, shorting and covering are simply a different kind of selling and buying, respectively). Also, only one trade can be made per time step, per trader.

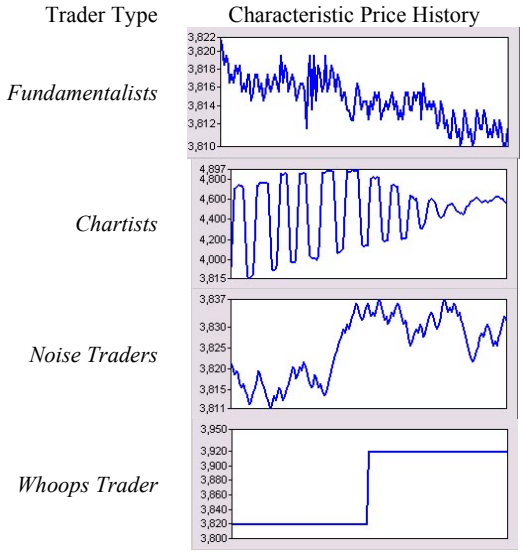


Figure 6.Summary of trader types and typical price histories. Top two include noise traders to trigger trading.

Fundamentalists. Fundamentalists calculate a "true value" of the underlying stock by averaging the price history a certain number of minutes back – a moving average of the form $T = \sum_{x=r}^{r-l} p_x$ where T is the true price, p is a historical price, r is the time of the most recently recorded historical price, and l is the number of time steps used in the computation. The trader then computes an upper threshold, U, and a lower threshold, L. Each is a percentage t away (50%=0.50) from the true price: $U = T(1+t)$ and $L = T(1-t)$. Once the price action has passed through one of the thresholds, a number of time steps must pass (greater than the fundamentalist's reaction time) before the fundamentalist is "awake" and able to trade. Once this reaction time threshold has been passed, the fundamentalist makes one trade per time step according to the rules:

Buy if $A \le L$ and a long position is not held

Sell if $B \ge T$ and a long position is held

Short if $B \ge U$ and a short position is not held

Cover if $A \le T$ and a short position is held

where B is the best bid, and A is the best ask. The trader holds a long position if he has bought, and he no long holds it if he sells; likewise for the short side. Characteristically, a fundamentalist appears to anchor the price history and dampen volatility.

Chartists. Chartists' trading is triggered by momentum – they trade in the direction of a trend when the trend is steep enough. The momentum M is computed: $M = (p_r / p_{r-l})/l$, where l is the number of time steps used in the calculation, p_r is the most recent historical price, and p_{r-l} is the price l time steps before p_r. A chartist's trading rules are:

If the number of my previous trades is less than the maximum allowed, then:

Buy if $M \geq p_r(1+T)$ and the trader does not hold a long position

Sell if $M \leq 0$ and the trader holds a short position *Short* if $B \geq U$ and a short position is not held

Short if $M \leq p_r(1-T)$ and the trader does not hold a short position

Cover if $M \geq 0$ and the trader holds a short position

where p_r is the most recent historical price and T is the threshold (5%=0.05). Buying, selling, shorting, and covering each count as one trade. Characteristically, chartists appear to reinforce trends and enhance both volatility and waves.

Noise Traders. Noise traders either buy or sell with equal probability in each time step. Noise traders are meant to reflect the apparent randomness in the real markets and act to move the price action around, in effect "tripping" the strategies of the other traders.

Whoops Trader. The whoops trader places one 10,000-stock order to buy at the market at 10:15am which represents the alleged trading mistake that happened on September 20, 2002 at 10:10am. While the trading mistake is reported to have occurred at 10:10am [19], orders on the London Stock Exchange may take many minutes before they are observed by other traders who then may take minutes more to act on that information. Because our synthetic traders immediately see information and react just as quickly, the simulation time of the mistaken order is adjusted to 10:15am. Figure 6 summarizes the various strategies and their characteristic price histories in isolation.

Genotype. The trading strategies of each trader are designed to make possible a wide variety of behaviors that can potentially produce the target pattern. The genotype of each simulation is, like experiment 1, composed of genes that are numbers. But in this experiment the numbers code for what are considered random variables – either how many of one kind of trader (the random variable is a constant in this case) or the α's and β's for a beta distribution. Random values drawn from the Beta distribution serve as specific parameters to traders. In this way, only α and β describe a distribution of values used to parameterize any number of traders, compressing the genotype. The Beta probability distribution has non-zero values and takes the form:

$$p(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

where x is real number in the interval **[**0,1**],** $p(x)$ is the probability at x, and Γ(n) is Euler's gamma function. We chose the beta distribution for our purposes because it is a bounded distribution that can take many shapes, it is nicely parameterized by just α and β, and it is easily scaled to cover each variable's legal bounds. Note also that a beta distribution with α=1 and β=1 is equivalent to a uniform distribution. Alphas and betas are each bounded between 1 and 100, and each parameter has an upper and lower bound for values drawn from this distribution which scales the corresponding beta distribution. Constants, α's, and β's comprise a genotype, and the parameter values drawn from these random variables and their bounds define the search space. The bounds and distributions for the parameters are defined as follows:

Fundamentalists: # of traders is a constant between 10 and 50; Trade size, Reaction time, Moving average length and threshold percentage are all drawn from Beta distributions within bounds [100,500], [1,50], [20,100] and [0%,5%], respectively.

Chartists: # of traders is a constant between 10 and 100; Trade size, Maximum # of trades, Momentum look back and Momentum threshold percentage are all drawn from Beta distributions within bounds [100,1000], [1,200], [1,5] and [0.1%,1%], respectively.

Noise traders: # of traders is a constant between 1 and 10; Trade size is drawn from a Beta distribution within bounds [100,200].

The genotype representation is summarized on Figure 7.

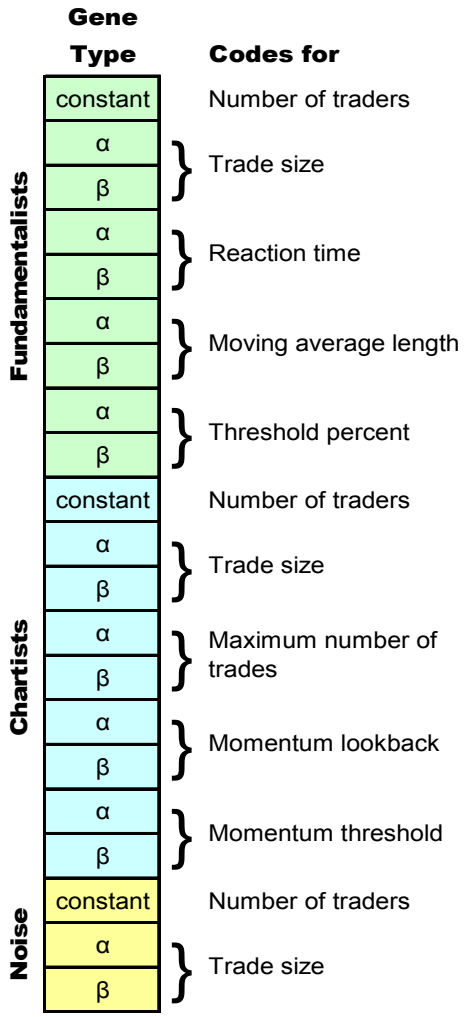


Figure 7. Summary of genotype representation.

Each simulation begins with the market maker quoting a best bid at $3820, best ask at $3821. All the market maker's orders are in lots of 100, and each simulation is run for the duration of the real dataset – from 9:00am to 11:30am in 1-minute increments.

B. Operators

Elitism was used the same as in Experiment 1.

Mutation occurs on each gene if rand<chance, where rand is a random number in the interval [0,1] and chance is the user-selected chance that a gene is mutated. New values for a given gene are chosen as follows: $v_n=min+rand*(max-min)$, where v_n is the new value, *min* is the lower bound for this gene, *max* is the upper bound for this gene, and *rand* is a random number in the interval [0,1].

Crossover is double-point crossover. The crossover itself happens identically to the crossover in experiment 1, except it crosses the statically-ordered genes of the experiment 2 genotype.

C. Experiment 2 Results

We were able to qualitatively match the target price history in an average of 10 generations, from a variety of starting populations. While "qualitative match" is an imprecise term, the characteristics we attempted to match are the amplitude, period, phase, and damping rate of the approximate wave – and of course the size, shape, and location of the price history. We matched reasonably well on each of these dimensions (Figure 8). The parameter values of the best solution are:

Noise traders

Number of traders (constant) = 8

Trade size (beta distrib.): α=97.02, β=49.75

Chartists

Number of traders (constant) = 27

Trade size (beta): α=12.68, β=40.16

Threshold % (beta): α=92.52, β=25.52

Look back (beta): α=93.64, β=30.77

Max trades (beta): α=41.28, β=73.28

Fundamentalists

Number of traders (constant) = 45

Trade size (beta): α=55.01, β=59.92

True price look back (beta): α=65.76, β=28.95

Percent away (beta): α=27.3, β=28.24

Reaction time (beta): α=99.35, β=83.8

It appears that chartists are responsible for the herding behavior (wave-form of the price action) and that nearly twice as many fundamentalists, submitting trades more than twice as big as chartists, are needed to dampen the effect of chartists. Fundamentalists use a roughly 60-minute moving average, have a ±2.5% response envelope, and might be considered patient in waiting a half hour after a threshold cross before awakening. The chartists seem to be conservative, waiting an average of 4 minutes before assessing a trend and acting only when that trend is relatively steep – that is, an average of 0.75% growth per time step (-0.75% in the case of a down-trend). Chartists having a maximum number of trades around 65 may or may not make a difference; the number of chartists who reached their maximum was not recorded. It might be the case that because none of the beta distributions were very close to the edges of their possible ranges, that reasonable bounds were chosen for the variables. The beta distributions also seem to be quite "tight," and we are tempted to conclude that narrow ranges tend to be most appropriate, but we are reminded that it is only when both α and β are very small that a beta distribution is not tight. It must be noted that while this best candidate appears to match that target exceptionally well, consider that some of the match is a result of random outcomes resulting from the noise trader and from parameters values being drawn randomly from beta distributions. Seeded with different random numbers, model simulations with the same parameters produce variety although they all exhibit similar qualitative features.

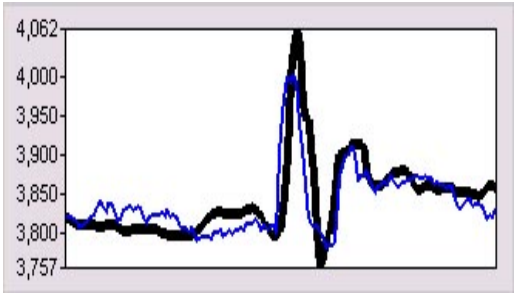


Figure 8. Best solution with data to be fitted (thick curve).

VI. Discussion

We showed with two simple financial markets examples how IEC can be used to perform agent-model inversion. When applied to simple models and reasonably small parameter spaces that have human-interpretable visualizations, IEC as a technique nicely combines human expertise with evolutionary computation and may improve the speed and accuracy of search in the fields of model inversion and model design, particularly when goal evaluation is multi-variate, highly complex, or qualitative. One useful extension to this work consists of using IEC to fit two or more aggregate-level, more complex, datasets simultaneously, thereby increasing the plausibility of the discovered model.

References

[1] Palmer, R. G., Arthur, W. B., Holland, J. H., Le Baron, B. & Tayler, P. (1994) Artificial economic life: a simple model of a stock market, *Physica D* **75**, 264-274.

[2] Arthur, W. B., Holland, J. H., LeBaron, B., Palmer, R. G. & Taylor, P. (1997) Asset pricing under endogenous

expectations in an artificial stock market, in *The Economy as a Complex Evolving System II*, eds. Arthur, W. B., Durlauf, S. & Lane, D. (Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Volume XXVII, Addison-Wesley, Reading, MA).

[3] Caldarelli, G., Marsili, M. & Zhang, Y.-C. (1997) A prototype model of stock exchange. *Europhys. Lett*. **40**, 479-484

[4] Levy, M., Levy, H. and Solomon, S. (2000). *Microscopic Simulation of Financial Markets: From Investor Behavior to Market Phenomena*. (Academic Press, San Diego, CA).

[5] Lux, T. & Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature* **397**, 498-500

[6] Mantegna, R. & Stanley, H. E. (1999). An Introduction to Econophysics, Cambridge University Press

[7] Bonabeau, E. (2002) Agent-based modeling: methods and techniques for simulating human systems, *Proc. Nat. Acad. Sci. USA* **99**, 7280-7287

[8] LeBaron, B. (2001) A builder's guide to agent-based financial markets. *Quantitative Finance* **1**, 254-261

[9] Epstein J. M., Axtell R. L. (1996) *Growing artificial societies: social science from the bottom up* (MIT Press, Cambridge, MA).

[10] Boschetti, F., Moresi, L. 2001. Interactive inversion in geosciences. *Geophysics* 66, 1226-1234.

[11] Wijns, C., Boschetti, F. & Moresi, L. 2003. Inversion in geology by interactive evolutionary computation. *Journal of Structural Geology,* in print.

[12] Dawkins. R. 1987. *The Blind Watchmaker*. W. W. Norton, New York.

[13] Sims, K. 1991. Artificial evolution for computer graphics. *Computer Graphics* **25**: 319-328.

[14] Sims, K. 1992. Interactive evolution of dynamical systems. Pages 171-178 in: *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (F. J. Varela & P. Bourgine, eds.), MIT Press, Cambridge, MA.

[15] Sims, K. 1993. Interactive evolution of equations for procedural models. *Vis. Comput*. **9**: 446-476.

[16] Banzhaf, W. (1997) Interactive evoution. In: Handbook of Evolutionary Computation (T. Baeck, D. Fogel, and Z. Michalewicz, eds), Ch. C2.10, pp. 1-5, Oxford University Press.

[17] Takagi, H. 2001. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* **89**: 1275-1296.

[18] Ball, P. (2002) Stock market shock explained. Physicists model recent trading frenzy. Nature Science Update, http://www.nature.com/nsu/020923/020923-18.html

[19] Muchnik, L. & Solomon, S. (2003). Statistical mechanics of conventional traders may lead to non-conventional market behavior. *Physica Scripta* **T106**, 41-47.