Economic Theory: Structural Abstraction or Behavioral Reduction?

Shyam Sunder

In physics, optimization is an organizing principle for natural phenomena. Entropy tends toward its maximum and marbles roll toward minimum potential energy, all without intent or purpose. Injection of this principle into economics initially followed the physicists' organizing perspective and helped develop the powerful insights of the abstract equilibrium theory. However, humans and their institutions being the unit of analysis, economists could not long resist the temptation to give optimization a behavioral spin. Photons may travel along paths that minimize their travel time without intention or purpose; but economists were all too human to think in a similar vein of the people buying ice cream or cars. Once optimization was posited as a behavioral principle of individual human beings, it was easy for cognitive sciences to show that it lacked descriptive validity. Individual behavior is more complex and less predictable. The aggregate characterizations of Walrasian abstraction could not be derived starting from such complex micro-level behavior. If psychology and equilibrium theory were to be reduced into a single science, something had to give. Given the cognitive limitations humans share with all organisms, validity and relevance of the conclusions of equilibrium theory became suspect.

The marriage of economics and computers led to a serendipitous discovery: there is no internal contradiction in suboptimal behavior of individuals

I am grateful for the comments from participants at the 2005 *HOPE* conference and the editors of this volume.

History of Political Economy 38 (annual suppl.) DOI 10.1215/00182702-2005-028

yielding aggregate-level outcomes derivable from assuming individual optimization. Individual behavior and aggregate outcomes are related but distinct phenomena. Science does not require integration of adjacent disciplines into a single logical structure. As the early-twentieth-century unity of science movement discovered, if we insist on reducing all sciences to a single integrated structure, we may have no science at all. In Herbert Simon's (1996, 16) words: "This skyhook-skyscraper construction of science from the roof down to the yet unconstructed foundations was possible because the behavior of the system at each level depended on only a very approximate, simplified, abstracted characterization of the system at the level next beneath. This is lucky; else the safety of bridges and airplanes might depend on the correctness of the 'Eightfold Way' of looking at elementary particles."

This is the story of how we found that economists can have their cake while psychologists eat it too. Willingness to abandon the reductionist agenda in social sciences—something natural sciences did in the early twentieth century—reveals that the predictive validity of Marshallian supply and demand theory need not depend on the theory's assumptions being literally descriptive of cognitively bounded human agents. The next section discusses the role computers have played in the design and operation of markets, and in modeling and facilitating the decisions of market participants. These developments, combined with the growth of experimental tradition in economics research and the use of such experiments for classroom instruction, set the stage for the discovery that structural properties of markets can be isolated from the behavioral patterns of their participants. The final section explores the scientific antecedents and consequences of this finding.

Computation and Markets

Each discipline is defined by, and changes with, its tools. Microscopes helped create new biology and chemistry, just as cloud and bubble chambers helped develop the physics of subatomic particles. In economics, mechanical computers of the first half of the twentieth century made the rise of econometrics possible. In the second half, electronic computers have gradually reshaped economics (Mirowski 2002). Interaction between economics and electronic computing and communications technology carries the potential to help shift the recent focus of economics from behavioral toward structural analysis.

Application of computers in trading can be divided into (1) organization and operation of markets and their associated information functions, and (2) decision support for, or as an agent to replace, traders in a market. Both these applications have been developed in parallel over the past four decades. In addition, the past two decades have seen a growing role for computation in developing a science and engineering of markets. I review the first two applications briefly as background to a discussion of the third—the main subject of interest here.

The early use of computers in markets focused on increasing the reliability of record keeping. Once the trading data were recorded, it was but a small step to use them to produce summaries, classification by various criteria, reports on trading activity, account books, and, soon thereafter, clearing and settlement. Computers made it possible to design market-clearing and settlement systems that could keep a tighter lid on exposure to risk inherent in trading. With the introduction of cheaper cathode-ray-tube (CRT) displays, periodic reports from line printers could be called up on-screen. As the database technologies advanced, the computer records of trading could be manipulated in real time with unprecedented flexibility to produce alternative tabulations relevant to the operators of market systems.

The real-time features made it possible for market regulators to use computers to monitor the actions of traders, and the behavior of prices and volume. It also made it possible for brokerage firms in stock and commodity markets to enforce the trading limits on their clients after considering their wealth and portfolios.

Computers and communications technologies were critical in increasing the geographic expanse of traders, whose orders could be channeled to markets without putting them at a large informational disadvantage with respect to the floor traders in stock and commodity exchanges. At least the completed transaction prices (but not the valuable bids, asks, and volume information yet) could be disseminated widely. In addition, these technologies brought unprecedented amounts of data, news, commentary, and research to market participants. It is difficult for a first-time visitor to the New York Stock Exchange not to be overwhelmed by the volume of continually changing information displayed on the surfaces that surround the people on the floor.

The availability of computer and communication technologies also created a demand for integration across markets; legislation and regulations for this purpose soon followed. This integration led to the demise

of many of the smaller local stock exchanges, as well as the creation of new electronic trading networks that had the freedom to offer more specialized services than those provided by the larger, established, regulated exchanges.

Computer as Trader

The use of computers initially to assist, and then to supplement, the trader grew in parallel with their use to operate markets. The extent to which the support process can proceed so computers can come close to substituting for human traders is, and will likely remain, controversial for both semantic as well as theoretical reasons. In the sense that the automobile supplements or substitutes for legs and the microphone for vocal chords, computers already perform as supplements or substitutes. Record keeping; access to market data, news, analysis, and research; communication of orders to the markets; computational models and aids to make trade decisions are important elements of the new system. CRT technology has made dynamic text and graphics displays of up-to-date information from various sources an indispensable part of a trading desk in most markets.

How well computing can replace the human trader remains controversial. On the one hand, the argument goes, because computer hardware and software are the creation of human intelligence, their capabilities must be limited by the state space conceived by their designer. When we become aware of new, heretofore unimagined, states of the world, we must redesign them to give them access to such states. Therefore, in a conceptual sense, human intelligence and worldviews constitute the outer limits of what computers can "see" and do. Their speed, large memory, and computational capabilities may enable them to be efficient assistants to traders, but they would never replace their human masters.

On the other hand, such a view of the abilities of computers as traders appears narrowly anthropomorphic and self-serving. Computing has enabled sciences to explore ideas that were otherwise inaccessible. Both memory and speed are important components of intelligence. To grant the dominance and flexibility of humans relative to computers in certain areas is not to concede the superiority of the former in all autonomous tasks. Computer chess programs reaching the competence of world champions of chess has not resolved this controversy, and they are not likely to do so.

Tooling Up for Computational Economics

CERL, Control Data, and PLATO

William C. Norris, the founder of Control Data Corporation, acquired the rights to the PLATO (Programmed Logic for Automatic Teaching Operations) system from its developer CERL (Computerized Education and Research Lab at the University of Illinois) and adapted it to Control Data's mainframe computer. PLATO was designed for K–12 and higher education, as well as military training and adult education (Norris 1997; Van Meer 2003). Economists' uses of computers to implement market institutions started with Arlington Williams (1980) programming double auctions on PLATO, initially to facilitate the conduct of laboratory experiments in economics with student subjects. Vernon Smith (1962) had conducted such experiments using paper, pencil, and chalkboard in a classroom.[1] Norris's PLATO was the initial platform to implement the oral double auction as a computerized double auction and soon led to the addition of various options and the proliferation of many different kinds of auctions on PLATO, largely to explore the properties of alternative auctions through experimental methods. This academic work preceded the introduction of e-commerce by almost fifteen years.

PLATO was a mainframe system implemented on specific Control Data computers. Smith's laboratory used leased phone lines to connect the terminals (CRT and keyboard) in Tucson, Arizona, with the computer located in Urbana-Champaign, Illinois. The popularization of a small, freestanding computer by Apple in 1978, followed by IBM's personal computer a few years later, changed that.

IBM Grant and a Market on PCs

Under IBM's grant of equipment, software, and support to the University of Minnesota's Project *Workscape*, I connected four IBM PCs and one PC-XT model in a small NetBIOS network to form a PC-based market in 1984–85 (Sunder 1987). By the late 1980s, the flexibility of PC networks led to a gradual replacement of the mainframe systems in major economics laboratories (e.g., Caltech, Iowa, and Arizona), before the introduction of the World Wide Web and e-commerce.

1. For an outline of the emergence of experimental tradition in economics, see Friedman and Sunder 1994, chap. 9.

With a small network of computerized PCs in my University of Minnesota laboratory, and software written to implement double auction and other simple markets for human traders using CRT screens and a keyboard, I had the toys at hand to start thinking about replacing human traders by automatons. In 1986–87 a small group that included Suk Sig Lim and Karim Jamal, doctoral students at the time, started meeting occasionally to discuss trading strategies in double auctions. These discussions were driven by several thoughts. Since a great deal of human double auction data were already available (from both oral and computerized auctions), a large part of this discussion focused on examining these data to identify algorithms that might capture the trading strategies used by individual traders (students in laboratory experiments).

The discussion also consisted of trying to design what might be reasonable algorithms for trading, given the profit goal, personal reservation values or costs, and any market data (bids, asks, and transactions) available to the trader in a double auction. In addition, if we could design automaton traders that approximated the behavior of human traders reasonably well, we could replace costly, inconvenient, and unreliable human traders by computer programs in our economics research experiments. Simon's work on goal-directed, adaptive, satisficing behavior was the only guide we could find to design such algorithms. Surprisingly, we could find little in economics or finance literature to help such a trader. These algorithms were not converted into computer code for trial in a market.

Program Trading as the Darth Vader of the Market Crash

After touching a high above 2722 in August 1987, the Dow Jones Industrial Stock Price Index dropped on 19 October from 2246 to 1738 (22.6 percent). The shock of the sharp drop in stock prices revived the fears of a repeat of the Crash of 1929 and led to multiple inquiries into the possible causes, including inquiries initiated by the New York Stock Exchange, the Securities and Exchange Commission (see U.S. Securities and Exchange Commission 1988), the U.S. Department of Treasury, and the Federal Reserve Bank's Board of Governors. A thread appeared to run through many of the reports of these inquiries made public in 1988—an accusing finger pointed at program trading and the small number of investment houses

that possessed the technological and financial sophistication to develop and use such techniques.[2]

A generic meaning of program trading should include the use of computer programs to assist or conduct trading in any significant manner. In the mid-eighties stock markets, the term had come to mean two specific activities—cross-market arbitrage and portfolio insurance, both involving trades in individual securities as well as in baskets of securities. Computers could be programmed to monitor the prices of individual securities and of the fixed baskets of these securities (indexes) often traded in the same or different markets. If the discrepancy among the prices was large enough (to cover the cost of transactions), the computer could be programmed to send buy or sell orders to the appropriate markets to earn profits through riskless arbitrage across markets.

In this application, the uninterrupted attentiveness of the computer in spotting the price discrepancies is crucial. Also important is the speed, so it can generate hundreds of buy and sell orders for traders to execute, to take advantage of the transient arbitrage opportunities before they disappear. Implementing portfolio insurance also depends on the same capabilities of the computer to issue a large number of buy or sell orders almost instantaneously whenever the market conditions fulfill prespecified criteria.

By 1987 a small number of elite investment houses had developed the technological capability to use computers for index arbitrage and portfolio insurance. These big boys could make money in a way that others could not. The computers could rapidly dispatch buy or sell orders for hundreds, even thousands, of different securities traded at different desks or in different markets. Individual traders could directly observe the action only at their own posts. Their past experiences suggested that, with the exception of major economy-wide events, the movement of prices of individual securities was largely independent. The simultaneous injection of, say, buy orders for one hundred major securities was an unfamiliar and largely invisible phenomenon that could move the market index in apparently mysterious ways. Science fiction imagery of evil and ever more powerful computers seems to have become a reality in security markets. Program trading was fast becoming a dirty word on the stock exchanges and was blamed for the Crash of 1987.

2. For an example of evidence to the contrary, see Harris, Sofianos, and Shapiro 1994. For a survey of the literature on program trading and market volatility, see Duffee, Kupiec, and White 1990.

A Course in Program Trading

Wall Street's common wisdom about the evils of program trading had the potential to stifle the promise of technology in the markets. I had moved to Carnegie Mellon in 1988. In early 1989, curious to learn about the consequences of program trading, I proposed to teach a new course on the subject to MBA students. There were no teaching or research materials available, and the term had hardly appeared yet in finance books or academic literature. Fortunately, in Carnegie Mellon's culture of academic entrepreneurship and experimentation with high technology, permission to offer a new elective in the fall of 1989 was secured with little more than a short paragraph for course description and a promise to think about what to teach in such a course and how to do it. Teaching was not distinguishable from learning.

Although the investment community used program trading in the narrow sense of index arbitrage and portfolio insurance described above, the course was to be a vehicle for exploring its broader meaning and consequences. The ideas and algorithms sketched in Minnesota in 1986–87 had to be tried out as automaton traders in a market. It was already clear that there was no obviously superior trading strategy in a double auction. To gain a better understanding of possible trading strategies, I needed to access the thinking of others on the topic. Members of the master's class, mostly MBA candidates and a few people from computer science, were to be the source of these ideas.

The course would ask the individual class members to develop their own algorithms. Each class session would consist of a tournament among such algorithms. The authors of the algorithms could watch the results of the double auction tournament live on the screen in graphic and text forms. After the session, the computer files of the results (behavior of their own, as well as others', algorithms, and the market outcomes) would be available to the class members. They could scrutinize the performance of their algorithms under competition and submit a revised algorithm for the next session. The time series of six algorithms generated by each class member during the course would serve as a library of ideas for further exploration of double auction trading. The research focus at this point was on the engineering of trading algorithms for double auctions, not on the engineering of market design or on the science of how markets behave. We already knew how markets behave, and why. At least so we thought.

Market 2001

In the summer of 1989 Dan K. Gode and I set out to build the platform for the computerized continuous double auction. The program-trading course was scheduled for the fall, which gave us three months to build and test the system. Fortunately, Gode had a strong engineering background and was enrolled in Carnegie Mellon's information systems PhD program.[3] A token-ring network of twenty-three IBM-PC 386 computers arranged inside a mobile trailer on partitioned desks, DOS operating system, and TurboPascal language constituted the platform.

The main elements of the Market 2001 system were the following:[4] (1) a manual trading mode with graphic/textual dynamic presentation of the market data on the screen in real time, (2) an equal access automaton trading mode with graphic/textual dynamic presentation of the market data on the screen for the authors of the automatons, (3) a mixed trading mode in which both automatons as well as humans could participate, (4) utilities for instructors to set up markets, (5) utilities to store intermediate results for fast retrieval by program traders during the trading session, (6) a command language in which participants could write the trading strategies, (7) a testing system in which participants could test the performance of their strategies against a specified background of other strategies on single machines (outside the laboratory), (8) storage of final results, (9) analytic tools for examining the data, and (10) user's manuals for instructors and students (Gode and Sunder 1989).

The course started with a few sessions of manual (i.e., students sitting in front of a computer screen and trading through keyboard and mouse) double auctions. The results were not surprising—they corresponded closely to Smith's (1962) report, which has been replicated innumerable times. After some initial volatility, prices settled down in close proximity of the point (or range) defined by the intersection of the initial market demand and supply.[5] In addition, the trades extracted virtually 100 percent of the

3. Gode, with his deep insights and vision to separate the important from the trivial, has been a close collaborator of mine in all aspects of this project since the summer of 1989.

4. The system was ambitiously named Market 2001, because it was thought that it would take a decade or so for security markets to catch up with this lab model to become fully open to automaton trading. The estimate turned out not to be ambitious enough; at the time of this writing, no security market is known to us to allow full access to trading automatons.

5. Following Smith 1962, laboratory double auctions are typically organized in discrete trading periods, with the endowment of the subjects refreshed at the beginning of each period,

maximum possible surplus that they could have extracted. It took the members of the class more time to learn the mechanics of trading (which keys to press to buy or sell, and what various numbers and charts on the screen show) than it took for the class market to arrive in close proximity of the equilibrium. These manual sessions gave the members a good understanding of the task for which they had to design an automaton in the course's next phase.

Outsourcing Yourself to a Code

The manual trading sessions were followed by tutorials (and individual assistance to class members) on how to use the software commands to convert their trading strategies into TurboPascal computer code (a subroutine that could run on one of the networked machines). Trading by their intuitions, students took less than a minute, often only a few seconds, to submit bids and asks and to start trading with little instruction beyond the mechanics of trading. Writing a computer code to do what they could easily do by sheer intuition, and little apparent thinking, turned out to be a challenging task for most of them. Students were asked to write down and submit a description of their strategy in plain English with the hope that this would be a useful intermediate step and help them make the transition from their thought to code. Despite coding assistance provided to class members, it was clear that the code submitted often did not match the intents and expectations of its creator. Students would mutter or scream at the computer when their codes failed to act in ways they expected during the class tournaments and were often dissatisfied with the poor profit performance in a competitive environment.

---

which is treated as a new game. Unlike Walrasian auction, in which all transactions take place at once at the same price, double auction transactions occur sequentially and often at different prices. Edward Chamberlin (1948) pointed out that each transaction changes the market demand and supply conditions under which the subsequent transactions occur. Therefore, in a double auction, it is useful to distinguish between the initial and instantaneous demand and supply as the trading proceeds. The meaning of demand and supply in a discrete-period laboratory implementation of double auctions is subject to alternative interpretations. In my viewpoint, for any defined market, demand and supply are flow variables with discrete stockasticity. Units entering the demand or supply sides of the market exit, either through transactions or through disappointment and exhaustion in completing a transaction. Both the standard static demand and supply of Marshallian analysis, as well as its dynamic interpretation by Chamberlin (1948), are different approximations of this reality and should not be taken as literal descriptions.

Teaching Is Learning

My first surprise came at the beginning of the first tournament session. It is easier to react to the actions of others in a double auction (to raise an existing bid by a small amount or to lower an existing ask) than to specify an optimal initial action. In a manual auction, it is rare that more than a few seconds elapse between the opening bell and the initial bid or ask submitted by someone in the market. Coding, however, is more formal, and it is more difficult to write in a code a decision on when to initiate bidding and to choose the level when there is no prior action to serve as a benchmark.

Following this logic, I had thought that the trading among programs might never get started (or take a long time before starting up) if everyone wrote a code that waits for the first move by others. This did not happen. The program-trading markets started just as promptly as the manual markets do, with one code submitting a bid of 1 (the minimum possible amount) and raising it by +1 in quick succession until it reached a certain level or was accepted in a transaction. The lack of knowledge of the logically best of all possible actions does not seem to keep traders from choosing an action in either manual or program-trading modes. Moreover, this tendency not to wait for the best possible action contributes to the efficacy of social institutions. Herbert Simon, the originator of the concept of bounded rationality and advocate of certain docility as a social value, would not have been surprised.

Efficiency as a Measure of Intelligence?

As the class proceeded, the participants gradually refined their strategies by adding contingencies and constraints, eliminated trading errors, and debugged their codes. On the first day of the class tournament, the traders collectively extracted only some 50–60 percent of the maximum possible (consumer plus producer) surplus. As the tournament proceeded, this percentage gradually crept up toward the nineties. Comparing these percentages to the almost 100 percent achieved in the manual session during the first meeting of the class, I used the allocative efficiency as an index of how the trading codes were getting better each week. The students also found it reassuring that their programs were getting "better" as measured by this aggregate metric. Assuming that better trading strategies on the part of the individuals must mean a larger fraction of the total surplus extracted in the market did not seem unreasonable at the time.

Code Length and Performance

Generally, the codes grew in length as the course proceeded. They varied from a page or two to more than fifty. There was little apparent relationship between the code's length and its profitability. While longer codes incorporated more contingencies and constraints, they were also more susceptible to programming bugs. Moreover, a continuous double auction put longer codes operating in a single-processor computer at a competitive disadvantage, at least in its Market 2001 implementation.

The double auction sends a continual stream of messages to the clients (including the confirmations and feedback on the actions submitted by the specific client, and the market information on bids, asks, and transactions is sent to all clients). This data stream keeps traders informed about the market's current status, which may change within a split second. As a single-processor machine receives the latest bits of information and considers its reaction to it, an elaborate code runs the risk of its reaction time being too long. The longer it takes to decide what to do in a given situation, the more likely it is that a change in the market's status may render the decision irrelevant. For example, suppose that a trader considers a bid of $30 that arrives at instant A, and decides at instant C to accept the bid. If another trader accepts the bid at instant B (before instant C), the shorter decision-processing time of the second trader puts the more elaborate code of the first at a competitive disadvantage. Humans may adjust their trading strategies to the tempo of trading intuitively. In the first generation of trading strategies observed in the program-trading class, such adjustments had not yet evolved.

In the meantime, the research agenda that motivated the course design was not going well.

Difficulty of Identifying Strategies from Code

As the library of coded trading strategies accumulated, Gode and I struggled to break down each strategy into separable elements, with the hope of isolating features that make a strategy more profitable. Once identified, such elements could be shown to the class as an example of how to write better trading strategies. Since theory provided little guidance, the class tournament could become a mechanism for discovering workable strategies and thus a source of learning about good trading strategies for the class and for the research community. At least that was the idea.

Implementing this idea turned out to be difficult. Trading codes needed to be broken down into modules or general ideas that could be interchanged with, or inserted into, other programs. As written by students, most of the code in a given strategy was too contingent on the other parts of the same code to permit easy identification of such modules. Under the pressure of the demands of the class, the task of analyzing the codes, rewriting them into modular form, and statistically comparing the performance of each code (seen as a bundle of distinct modules) with the hope of identifying the efficiency of each module slipped to the back burner.

What Is a Better Trading Strategy?

A second problem—about the feasibility of outrunning one's own shadow—gradually rose to the level of consciousness. Suppose we had succeeded in breaking down the trading programs into perfectly interchangeable modules, and statistical analysis helped us assign a profitability score to each module. Suppose we gave this information to participants in the class or to a group of experts to use in creating their own respective trading codes for the next tournament. Would this process of diffusion and competition simply reduce the most profitable trading ideas to be mere averages, without adding anything to the overall profitability? Isn't the profitability of a trading program a relative, not absolute, property? Perhaps it is not a property of individual programs and is associated only with a constellation of programs present in a market. These doubts, too, had to be put on the back burner, as the students pressed me to teach them how to write winning trading strategies.

The master's students (mostly MBAs) had enrolled in the class with the hope of learning to write winning strategies in a trading game. Their learning came from the opportunity to implement their own hunches of what might work in the trading environment, the programming language, and the testing tools created for the course. The environment of the class had their competitive juices flowing so they could try out their own best ideas and compare them with the ideas generated by others in the class.[6] As intense, frustrating, and fulfilling as this experience was, the class members seemed to assume that I had better strategies up my sleeve. After all, teaching students how to do things better is the foundation of postwar management science. At the very least, they wanted to have the chance to

6. At the end of the class, the competition over, most students volunteered to share their codes.

have their trading strategies go head-to-head with the instructor's. For an instructor without a trading strategy, this demand presented a problem.

Most university courses cover disciplines in which the instructor has some expertise, and the students expect to gain from this expertise through lectures, class discussions, readings, assignments, and examinations. While almost all of their learning comes from their own efforts in the form of the last four elements, students tend to underrate the instructor as an impresario of their learning environment and overrate him or her as a dispenser of knowledge through lectures. The program-trading course had been billed as one of experiential learning, in which a special learning environment had been created in the laboratory. Even though I had little special knowledge of trading strategies to convey to the class, such a learning environment was unique and pedagogically justified. Yet the presumption that I must know more than the students about profitable trading strategies was hardly justified.

There was, and is, no known "best" strategy for trading in a simple double auction. By this time, Gode and I knew that some short and simple trading strategies did rather well in the market, not because they were smart in the sense of being sophisticated but because they were "there," all the time, and were fast. Being there, they say, is half the winning. What could be the simplest "being there" strategy? Generate proposals that would not lose money. By this time, we also knew that the price improvement rule (any bid must be higher than an existing bid in the market and any ask must be lower than an existing ask in the market in order to be valid) serves as a safety net for traders. On a hunch, we wrote a simple, one-line code late one September night in the trailer lab and tried it in a market of twelve clones of the automaton (six assigned the role of buyers and six as sellers).

This Makes No Sense

The transaction prices bounced around over a wide, though gradually narrowing, range as the first period of trading ended. The same thing happened in each of the six (statistically identical) periods we tried. The automaton markets ran too fast (some twenty to thirty transactions in a fifteen-second period) to notice anything more than that. When we reviewed the data on the graphic screen in slow motion (with the *review* utility), we noticed that the last few transactions of every period were close to the Walrasian equilibrium price. This made no sense to us.

The one-line code of the trading programs continually generated bids (for buyer role) or asks (for seller role) that were uniformly distributed random numbers drawn from the largest possible range (with an arbitrary upper limit for sellers) over which they would not lose money. In other words, buyers picked random numbers between zero and their value and submitted them as bids; the sellers picked random numbers between their cost and a fixed upper limit and submitted them as asks. There was no obvious reason why a market populated by such traders would generate prices so close to equilibrium. There must be a programming error somewhere. It was getting close to midnight, and we called it a day.

The next day, we repeated the market from the night before, and the last few transactions still occurred at prices close to the predicted equilibrium.[7] We changed the demand and supply functions, and the trader endowments, and saw similar results. The shock came when we examined the allocative efficiency of these markets—it was always close to 100 percent. This trading program had no memory, learning, adaptation, maximization, or other human faculties to which intelligence is often attributed. Maximizing the behavior of individuals had been drilled into us as the very foundation of economics. We had been using the gradually increasing allocative efficiency of the tournament markets as a barometer of how the intelligence of the participant programs was growing with each passing week. Yet a market consisting entirely of single-line automatons not only converged (albeit noisily) to equilibrium price but also achieved almost perfect allocative efficiency.

Something must be amiss. After months of intensive development and debugging, our system and the programs were always the first suspect whenever we observed anything unusual. It was time for another thorough review of the system. We could find no errors. Every market we ran yielded qualitatively similar results. How could we make sense of them? What could these data be telling us? We sensed we had something interesting, but did not know what it was.

Market 2001 was a complex system that coordinated twenty-four networked computers simultaneously through a single program running on

7. Fortunately, automaton markets took only some ten or twenty seconds to do one period of trading. If our graphic-intensive program had been less demanding on the Intel 386 machines of the time, each period would have taken no more than a second or two to run, a huge savings in time, money, and convenience over markets run with human traders. More important, human traders choose their own behavior, but we get to choose how the programs "behave."

a server. This was an ideal configuration, with each automaton trader running independently on its own processor, making trading decisions unencumbered by demands on the processor from other users. However, we did not fully understand how the lower layers of the IBM token-ring network functioned. Despite the supposed equal access to the token on the ring of computers, we had noticed what appeared to be some persistence in client computers that received the token after it was released. The first thing we had to confirm was that the data we had observed were not an artifact of how the network functioned and could be replicated with all automaton traders running with randomized access on the same processor as the market control program. This is much easier to do than writing and running programs on a network. The network results were replicated on a single-processor machine.[8]

Next we tried to analyze a model of a market populated by simple automatons we had used in the lab to see if we could obtain similar results. As the statistical analysis began to yield high allocative efficiencies (Gode and Sunder 1993b, 1997), the idea that efficiency, in its first order of magnitude, may be the property of the market institution, largely divorced from individual behavior, also began to take shape. We had merely stumbled on the result when not looking, but were lucky enough to recognize it as we got up to dust ourselves off.[9]

Since the students wanted to compete against the instructor's program, we set up a market in which all members of the class were assigned the role of manual buyers, to trade against an equal number of copies of our one-line program, which acted as sellers. The class was informed how the seller program worked (submitting randomly drawn numbers above their costs as asks). Its very simplicity was to be the instructor's protective cover and nervous joke, when it would lose to the student traders. We slowed the speed of the seller programs to one ask every two seconds to make it comparable to the speed at which humans can enter their bids through the keyboard. The simple programmed sellers rapidly lowered the market ask until, close to the supply function, it could go no lower and most trades occurred at prices just above the seller costs. Allocatively, the market was

8. Single-processor implementation of continuous double auction requires some additional specification of how the processor time is distributed among the traders and the server, and how the bids and offers are processed. These details are given in Gode and Sunder 1993b, but are not relevant to the present discussion.

9. See Gode and Sunder 1993a, 1994, 1997, and 2004; Jamal and Sunder 1996, 2001; Bosch-Domènech and Sunder 2000; and Gode, Spear, and Sunder 2004.

highly efficient, but virtually all the surplus ended up in the hands of the students, and the simple program traders made little profit. The students knew they had done well by beating the program traders. There was joy in the class as the students exchanged high fives.

In the following period, as the program traders quickly lowered the market ask, each student had to trade off between waiting for the asks to drop even further and picking the low-hanging fruit before someone else took it. Students realized that it was in their collective best interest to wait till the market asks could drop no further before concluding a transaction. They exhorted one another to wait, but had no way of coordinating a collective decision and ended up competing against one another. Consequently, the second-period transaction prices exceeded the first-period prices, and the third-period prices exceeded the second-period prices. In the following periods, the prices settled to a level below the equilibrium, giving students a significant edge over the program traders. Yet the students had realized that while they learned, their strategizing had shifted the terms of trade against them and in favor of the "zero-intelligence" traders (as we had christened them), who had no ability to anticipate or strategize.

Computers had opened a window into aspects of economics we could not access before, and revealed new secrets. Ironically, it was not the celebrated optimization capabilities of computers that brought this about. Instead, it was the use of the computers to isolate and examine the performance of markets populated by traders of arbitrarily chosen properties. Computers served as an instrument to deconstruct the complexity of human behavior. Isolation of elements of behavior made it possible to identify the characteristics of aggregate outcomes that may or may not be linked to individual behavior.

Optimization Principle: From Physics and Biology to Economics

When a marble rolls down the side of a bowl and comes to rest at the bottom, physicists know the marble minimizes its potential energy. When a photon leaves the sun and travels to the eye of a fish swimming underwater on earth, the physicist knows that the photon bends just sufficiently at the surface of water so its total travel time from the sun to the fish's eye is minimized. How does the marble decide where to go and where to stop? How does the photon know where to turn and by how much? Why

do they care to minimize or maximize anything? These are not meaningful questions to a physicist. In physics optimization is used as a fundamental organizing principle of nature. Minima or maxima are guides to how physical systems behave. In biology:

> At multiple hierarchical levels—brain, ganglion, and individual cell—physical placement of neural components appears consistent with a single, simple goal: minimize cost of connections among the components. The most dramatic instance of this "save wire" organizing principle is reported for adjacencies among ganglia in the nematode nervous system; among about 40,000,000 alternative layout orderings, the actual ganglion placement in fact requires the least total connection length. In addition, evidence supports a component placement optimization hypothesis for positioning of individual neurons in the nematode, and also for positioning of mammalian cortical areas. (Cherniak 1994, 2418)

The objects of analysis in physics—marbles and photons—are inanimate. We do not ascribe intentionality to them. Physicists talk about the behavior of these elements only in the sense that they follow the immutable laws of nature the physicists seek to identify. Even in the passage quoted above from biology, two out of the three objects of analysis are ganglion and individual cells, which are physical objects with no ascribable intentionality. Their behavior, too, is supposed to follow the immutable laws of nature the biologist seeks to identify. Physicists, biologists, chemists, and many other scientists can talk about the behavior of objects and use optimization as a structural principle to gain an understanding of the big picture. Structural models are about the proverbial forests, not the trees; they concern the existence, type, and growth of the forest, not the location and height of individual trees. They shield us from getting lost in the detail.

Economics appears to have borrowed the optimization principle from physics (and increasingly from biology, in recent decades). Instead of inanimate marbles or photons, we and the institutions in which we live and work are the objects of economic analysis. When applied to our own self-conscious selves, "behavior" takes on the burden of intentionality not necessary or present in the natural sciences (Sunder 2004).

On the one hand, we proudly claim to have a unique attribute of free will, placing how we act beyond the kinds of laws physicists devise to describe the behavior of marbles and photons. Yet social sciences try to

identify general laws that may help us understand, explain, and predict human actions as if we were some not-so-special kind of marble.

It was not surprising that in borrowing the concept of optimization from physics, economists transformed it from a structural or organizational principle to a behavioral principle.[10] Optimization was to be regarded as a matter of conscious choice of the best of the known and available options for individuals. Cognitive psychology soon made it clear that, when acting by our intuition, we humans are not good at optimization (Simon 1957). If we were, we could have no free will. We could not have both.

Walrasian Abstraction or Behavioral Reduction

The consequences of this dilemma between optimization and free will pervade twentieth-century economics. On the one hand, adoption of optimization as a behavioral principle has made optimization the defining characteristic of economics and economic behavior. Findings of cognitive sciences have therefore forced an increasing number of economists to look at the optimization principle with a skeptical eye, if not abandon it altogether. John Conlisk (1996), for example, gives four reasons for dropping the "infinite in faculties" assumption in favor of incorporating bounded rationality in economic models: empirical evidence on the importance of bounded rationality, proven track record of bounded rationality models (in explaining individual behavior), unconvincing logic of assuming unbounded rationality, and the cost of deliberating on an economic decision. Again, the focus is on the "trees," not the "forest."

Economics can be usefully studied as an abstraction of certain social phenomena at the level of a collectivity of individuals. All sciences must make some assumptions about the phenomena at the level of details they do not wish to delve into. Reluctance to make such assumptions leads one to the infinite regress of the "unity of science" movement of the early twentieth century (Neurath, Carnap, and Morris 1969). Insistence that all such assumptions, independent of their specific roles in theory, be descriptively valid creates a burden that is both unreasonable as well as unproductive.

10. Avinash Dixit (1990, 1): "Economics has been defined as the study of making the best use of scarce resources, that is, of maximization subject to constraints. The criteria being maximized, and the constraints being imposed on the choice, vary from one context to the next: households' consumption and labor supply, firms' production, and governments' policies. But all constrained maximization problems have a common mathematical structure, which in turn generates a common economic intuition for them."

When we study aggregate economic behavior in markets, especially their allocative efficiency, assuming that individual traders optimize their decisions makes it easier to derive equilibrium outcomes. However, the optimization assumption, too, is largely a matter of convenience. The behavior of markets populated by zero-intelligence traders suggests that the markets have structural properties of their own. Economics, like physics, can usefully focus attention to these structural properties, instead of getting distracted by the details of individual behavior. Psychology—a discipline at least as old and deep as economics—is better equipped for that purpose.

References

Bosch-Domènech, Antoni, and Shyam Sunder. 2000. Tracking the Invisible Hand: Convergence of Double Auctions to Competitive Equilibrium. *Computational Economics* 16.3:257–84.

Chamberlin, Edward H. 1948. An Experimental Imperfect Market. *Journal of Political Economy* 56 (April): 95–108.

Cherniak, C. 1994. Component Placement Optimization in the Brain. *Journal of Neuroscience* 14.4:2418–27.

Conlisk, John. 1996. Why Bounded Rationality? *Journal of Economic Literature* 34 (June): 669–700.

Dixit, Avinash K. 1990. *Optimization in Economic Theory*. London: Oxford University Press.

Duffee, G., P. Kupiec, and A. P. White. 1990. A Primer on Program Trading and Stock Price Volatility: A Survey of the Issues and the Evidence. Working Paper, Board of Governors of the Federal Reserve System.

Friedman, Daniel, and Shyam Sunder. 1994. *Experimental Methods: A Primer for Economists*. Cambridge: Cambridge University Press.

Gode, Dhananjay K., and Shyam Sunder. 1989. Program Trading Manual. Unpublished manuscript, Carnegie Mellon University.

——. 1993a. Allocative Efficiency of Markets with Zero Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* 101 (February): 119–37.

——. 1993b. Lower Bounds for Efficiency of Surplus Extraction in Double Auctions. In *The Double Auction Market: Institutions, Theories, and Laboratory Evidence*, edited by Daniel Friedman and John Rust. Santa Fe Institute Studies in the Sciences of Complexity. New York: Addison-Wesley.

——. 1994. Human and Artificially Intelligent Traders in a Double Auction Market: Experimental Evidence. In *Computational Organization Theory*, edited by Kathleen Carley and Michael Prietula. Hillsdale, N.J.: Erlbaum.

——. 1997. What Makes Markets Allocationally Efficient? *Quarterly Journal of Economics* 112.2:603–30.

——. 2004. Double Auction Dynamics: Structural Effects of Non-Binding Price Controls. *Journal of Economic Dynamics and Control* 28.9:1707–31.

Gode, Dhananjay K., Stephen Spear, and Shyam Sunder. 2004. Convergence of Double Auctions to Pareto Optimal Allocations in Edgeworth Box. Yale ICF working paper no. 04-30.

Harris, Lawrence, George Sofianos, and James E. Shapiro. 1994. Program Trading and Intraday Volatility. *Review of Financial Studies* 7.4:653–85.

Jamal, Karim, and Shyam Sunder. 1996. Bayesian Equilibrium in Double Auctions Populated by Biased Heuristic Traders. *Journal of Economic Behavior and Organization* 31.2:273–91.

——. 2001. Why Do Biased Heuristics Approximate Bayes Rule in Double Auctions? *Journal of Economic Behavior and Organization* 46.4:431–35.

Mirowski, Philip. 2002. *Machine Dreams: Economics Becomes a Cyborg Science*. Cambridge: Cambridge University Press.

Neurath, Otto, Rudolf Carnap, and Charles Morris, eds. 1969. *Foundations of the Unity of Science: Toward an International Encyclopedia of United Science*. 2 vols. Reprint, Chicago: University of Chicago Press.

Norris, William C. 1997. PLATO Pioneered Widespread Improvements in Education and Training. *Legacy: Newsletter of William C. Norris Institute*. September. http://www.stthomas.edu/norrisinstitute/PLATO.pdf.

Simon, Herbert A. 1957. *Models of Man: Social and Rational*. New York: Wiley.

——. 1996. *The Sciences of the Artificial*. 3rd ed. Cambridge, Mass.: MIT Press.

Smith, Vernon L. 1962. An Experimental Study of Competitive Market Behavior. *Journal of Political Economy* 70 (April): 111–37.

Sunder, Shyam. 1987. Systems Development for an Experimental Economics Laboratory. In *Proceedings of the IBM ACB University Conference*, 505–12. Boston.

——. 2004. Markets as Artifacts: Aggregate Efficiency from Zero-Intelligence Traders. In *Models of a Man: Essays in Memory of Herbert A. Simon*, edited by M. E. Augier and J. G. March. Cambridge, Mass.: MIT Press.

U.S. Securities and Exchange Commission. 1988. *The October 1987 Market Break*. SEC staff report. Chicago: Commerce Clearing House.

Van Meer, Elisabeth. 2003. PLATO: From Computer-Based Education to Corporate Social Responsibility. www.cbi.umn.edu/iterations/vanmeer.html.

Williams, Arlington W. 1980. Computerized Double-Auction Markets: Some Initial Experimental Results. *Journal of Business* 53 (July): 235–58.