Dealing with the Complexity of Economic Calculations

Professor John Rust
Yale University[1]
Revised, October, 1997

Abstract: This essay is a response to a growing negative literature that suggests that neoclassical economic theories based on hypotheses of rationality and equilibrium are of limited practical relevance because they require an infeasibly large number of calculations. Many of the negative results are translations of abstract complexity bounds from the computer science literature. I show that these bounds do do not constitute proofs that difficult economic calculations are "impossible" and discuss the type of hardware and software that can make it possible to solve very hard problems. I discuss four different ways to break the curse of dimensionality of economic problems: 1) by exploiting special structure, 2) by decomposition, 3) by randomization, and 4) by taking advantage of "knowledge capital." However these four methods may not be enough. I offer some speculations on the role of decentralization for harnessing the power of massively parallel processors. I conjecture that decentralization is an efficient "operating system" for organizing large scale computations on massively parallel systems. Economies, immune systems and brains are all types of massively parallel processors that use decentralization to solve difficult computational problems. However knowledge capital, in the form of effective institutions, is necessary to ensure that decentralization leads to effective cooperation rather than anarchy and chaos. I suggest that one reason why economists have had great difficulty computing approximate solutions to detailed models of individual behavior and large scale models of the economy is that they are not using appropriate hardware and software. Economists should structure their computations to mimic the key operating features of brains and economies, using parallel processing and decentralized 'agent-based" modeling strategies to solve more realistic economic models where solutions arise endogenously as "emergent computations".

[1] Invited paper for workshop on "Fundamental Limits to Knowledge in Economics", Santa Fe Institute, July 31-August 3, 1996. Financial support from the Alfred Sloan Foundation is gratefully acknowledged. This draft benefited from comments from Albert Ando, Kenneth Arrow, Robert Axtell, Dirk Bergeman, Samuel Bowles, Steven Durlauf, Eric Friedman, John Geanakoplos, Stuart Kauffman, Charles Manski, Shailendra Raj Mehta, Danny Quah, Ariel Rubinstein, Peter Tinsley, Joseph Traub, Michael Wellman, and Henryk Woźniakowski.

1. Introduction

Are there fundamental limits to knowledge in economics? Countering a distinguished tradition of impossibility theorems in economics, this essay is an attempt to suggest positive answers to two different aspects of what might be called the "problem of economic complexity":

1. does the computational complexity of economic calculations (i.e. those required to implement various economic concepts such as optimization, equilibrium, etc.) place inherent limits on the ability of *economists* to model the behavior of economies and economic agents?

2. does the computational complexity of economic calculations place inherent limits on the ability of *economic agents* to behave according to existing economic theories of optimization and equilibrium?

A negative answer to question 1, i.e. that "most" economic problems are computationally intractable, would seem to force us to the conclusion that efforts of economists such as Guy Orcutt to build a tractable, realistic, and empirically estimable model of the economy from microfoundations have been futile and misdirected.[2] Instead, the negative view suggests that economic science cannot hope to provide much more than a disjoint collection of highly simplified "toy models" which are either sufficiently stylized to be solved analytically, or are simple enough to be approximately solved on digital computers. A negative answer to question 2 would have equally serious implications for economic science, since it would imply that most existing economic theories are irrelevant and even misguided as positive models of the economy and individual economic behavior. Specifically, if there is an inherent "curse of dimensionality" underlying most economic calculations, the realization that actual economic agents have limited computational capabilities and operate in very complex environments with extremely high dimensional state spaces and choice sets, would appear to force us to abandon the standard "rationality hypothesis" that economic agents behave "as if" they were making the vast number of computations implied by existing economic theories.

Section 2 reviews some key results from the literature on computational complexity which constitute the logical underpinning for the argument that difficult economic calculations are "impossible". The complexity bounds have been used to show that many economic problems ranging from computing a Walrasian equilibrium to an economy or a Nash equilibrium to a game, to

[2] Orcutt lead a team of researchers who failed in an ambitious attempt to build a "bottom up" model of the U.S. economy at the Social Systems Research Institute at the University of Wisconsin in the early 1960s. Subsequent attempts to model the economy have used highly aggregative nonlinear economic models which do not attempt to deduce the time series properties of macro variables via aggregation of behaviorally motivated microrelations.

solving a stochastic team decision or dynamic programming problem, are either *non-computable* or *intractable.*

Section 3 describes hardware and software "solutions" to the problem of economic complexity. First, I show that the fact that an economic problem is computationally intractable does not necessarily imply that it is impossible: it is an empirical question whether the available hardware is able to compute a sufficiently precise approximate solution to a specific problem in the available amount of time. Second, I argue that worst case complexity bounds can be highly misleading by showing that many supposedly intractable problems become tractable when we consider alternative measures of complexity which account for different amounts of prior information about a problem and allow for different ways of assessing the accuracy or quality of an approximate solution. I discuss four different ways to break the curse of dimensionality of economic problems: 1) by exploiting special structure, 2) by decomposition, 3) by randomization, and 4) by taking advantage of "knowledge capital."

In my discussion of the importance of knowledge capital I suggest that long term evolutionary forces have enabled nature to "discover" powerful hardware and software that is capable of solving extremely difficult large-scale computational problems that we are presently unable to solve. We are only just beginning to appreciate and understand the methods nature uses to solve hard problems. Examples include the immune system that is able distinguish friend from foe and destroy myriads of potential pathogens, and the human brain that is able to solve many highly complex problems such as image recognition, language acquisition, and many other classification and decision problems. While these systems are by no means perfect (i.e. there are viruses that the immune system cannot destroy and there are many mathematical problems that most humans are unable to solve), their ability to behave sensibly and perform a wide range of difficult tasks well in a constantly changing environment is truly remarkable. Although researchers in economics, computer science, and artificial intelligence have made quite a bit of progress in the last 30 years, with few exceptions we are still unable to write computer programs that are competent in performing even relatively mundane tasks in highly controlled and simplified environments (e.g. navigating through obstacles, stacking blocks, recognizing faces, understanding spoken language, etc.). I believe that efforts to develop better hardware and software for solving complex economic problems will be accelerated if we try to discover and emulate the methods that nature uses to "solve" hard problems.

In section 4 I offer some conjectures about the some of the methods nature uses to "solve" hard problems. While I certainly don't pretend to have a complete answer to this question, it does seem clear that most of the powerful information processing systems that we see in nature share two key features: *1) massive parallelism,* and *2) decentralization.* Massive parallelism means that the system contains thousands, millions, or even billions of relatively homogeneous processors or *agents*. Decentralization means that there is no identifiable “central planner” that controls the individual agents in the system. Instead, the agents appear to operate autonomously according to their own preferences or objective function, perhaps influenced by messages, competition or other types of interactions with other agents. However when we observe the collective outcome of the agents’ interactions it appears as if they had cooperated to solve a well-defined overall problem. Adam Smith was perhaps one of the first to note this feature of decentralized systems in economic contexts when he described the economy as operating as if it were guided by an invisible hand. Paralleling theorems that decentralization of economic resource allocation processes leads to outcomes which are allocationally efficient (e.g. the First Theorem of Welfar economics) and informationally efficient (e.g. Mount and Reiter’s 1974 Theorem on the size of message spaces), I conjecture that decentralization of massively parallel computing devices leads to outcomes which are in some sense “computationally efficient.” The argument is that ultimately infeasible to adopt a centralized approach to computation in massively parallel systems sufficiently large number of processors, analogous in many respects to the argument that it is ultimately infeasible to adopt a centralized approach to resource allocation in large economies.

2. Computational Complexity and Limits to Knowledge in Economics

There is already a fairly large literature of formal and informal arguments that rational and equilibrium behavior is “impossible” and a parallel literature on the existence of fundamental limits to knowledge in economics. A number of informal “impossibility” arguments were proposed in response to increasingly elaborate game theoretic and rational expectations theories that were developed in the 1970’s. These arguments can be caricatured as follows: “If the brightest economists are unable to find exact analytic solutions to their models or require supercomputers to compute approximate numerical solutions to their models, how can they expect the ‘man on the street’ to behave according to their theories?”. It is relatively easy to dismiss these sorts of arguments since they depend on the implicit assumption that the intelligence of the “man on the street” is quite limited while the power of supercomputers and the intelligence of the best economists is nearly unbounded. The difficulties researchers in economics and artificial intelligence have encountered in developing algorithms to perform even simple tasks that the “man on the street” can do effortlessly and instantaneously suggests that the true state of affairs is more nearly the opposite of what the informal arguments assume. The rest of this section focuses on formal impossibility arguments since these arguments are much harder to dismiss.

To my knowledge von Mises and von Hayek provided some of the first formal arguments for fundamental limits to knowledge in economics. In his 1974 Nobel Memorial Lecture, "The Pretence of Knowledge" Hayek presented the now well-known argument of the impossibility of central planning based on the assumption that it is impossible to observe and transmit all of the information necessary to compute an equilibrium or social planning solution for the economy. Interestingly, he did not appear to question economists' ability to actually calculate an optimal allocation if it were somehow possible to transmit all the "facts" of the economy to a central planner:

> "A theory of essentially complex phenomena must refer to a large number of particular facts; and to derive a prediction from it, or to test it, we have to ascertain all these particular facts. Once we succeeded in this there should be no particular difficulty about deriving testable predictions — with the help of modern computers it should be easy enough to insert these data into the appropriate blanks of the theoretical formulae and to derive a prediction. The real difficulty, to the solution of which science has little to contribute, and which is sometimes indeed insoluble, consists in the ascertainment of the particular facts." (von Hayek, 1989, pp. 6–7)

A similarly optimistic view of the power of computers was espoused by Lange, one of the chief proponents of the efficacy of centralized planning, in response to criticisms of his approach by Hayek and others: "My answer to Hayek and Robbins would be: so what's the trouble? Let us put the simultaneous equations on an electronic computer and we shall obtain the solution in less than a second." (Lange, 1967, p. 158).

Although the main message of this essay is that computational problems are not insurmountable, I would not be so bold or naîve as to claim they are trivial. While I certainly agree with Hayek's argument that there are limits to knowledge due to the fact that information is decentralized, the primary focus of this essay is on the limits to knowledge arising from the issue of whether the required computations are too hard. The remainder of this section summarizes the key "limits" arguments that have been based on various versions of the theory of *computational complexity.* Roughly speaking, these arguments state that there are inherent limits on the ability of digital computers to compute exact or even approximate solutions to economic problems. Since these computational limits apply to any conceivable hardware or software, then to the extent we believe the human brain is a highly elaborate "chemical computer", the complexity bounds imply that there are both "fundamental limits to knowledge" and "fundamental limits to rationality" in economics.[3]

---

[3] The hypothesis that human brains are subject to the same fundamental limitations as digital computers is of course the subject of a long, hotly contested and still unresolved "mind-brain" debate in the philosophy literature. For an accessible synopsis of this debate, see Anderson (1964). See also Penrose (1989), who argues that there is an essential non-algorithmic component to conscious thought processes in the sense that phenomena such as creativity and consciousness may be a result of quantum-mechanical effects within the brain, but cannot be explained by modeling the human mind as a highly elaborate type of "Turing machine".

Economists have appealed to computational complexity theory at several levels in their attempts to establish fundamental limits to knowledge in economics. The most basic results have appealed to the theory of *effective computability,* imposing the very minimal requirement that any economic problem that we are interested in solving be *effectively computable:* i.e. there exists an algorithm that is guaranteed to terminate with the correct solution to the problem in a finite amount of time when run on any standard finite state computer (i.e. a Turing machine). This is an extraordinarily weak requirement because it only requires the algorithm to *eventually* return the correct answer: there is no requirement that the algorithm produce an answer in a realistic amount of time using currently available hardware. A classic example of a problem that is not effectively computable is the *halting problem.* Roughly speaking, there is no effective method for determining whether a given computer program running on a given input will eventually halt.[4]

To my knowledge the earliest translation of impossibility results in logic and computer science and logic to problems economics was due to Rabin (1957) who proved that there exist games whose equilibrium strategies are not effectively computable. Binmore (1987) established a similar result, noting "None of this is at all profound. Mathematically, all that is involved is a trivial adaptation of the standard argument for the halting problem for Turing machines." (p. 176). It isn't clear whether these results have anything to say about the narrower and more relevant issue of whether rational behavior is possible in *practical contexts* since they rely on non-constructive arguments to establish the existence of games whose equilibria are not effectively computable. The relevant question is whether equilibria are effectively computable *for the games economic agents actually play.*

Nachbar (1993) partially addresses this issue by proving a stronger impossibility result. He showed that in infinitely repeated two player discounted games, the problem of determining a best response to any *fixed* strategy of an opponent is not effectively computable in games where the discount factor is sufficiently close to 1. Nachbar concludes that his result "casts doubt as to whether it makes sense to model 'fully rational' decision making via Turing machines." (p. 2). Lewis (1985, 1986, 1992) has proven that in addition to N-person games many other key economic concepts such as individual demand correspondences, Walrasian general equilibria, Stackelberg equilibria, and Hurwicizian resource allocation problems are not effectively computable, even when we restrict attention to the class of problems where the descriptions of primitives of the problem (endowments, preferences, choice sets, etc.) are *recursively presentable* i.e. all functions

[4] A related result is Gödel's Incompleteness Theorem, which states that there is no formal system (i.e. computerized theorem-proving program) that is capable of proving all true statements in any axiomatic system of sufficient richness (such as arithmetic). That is, one can prove that there exist theorems of arithmetic which are true but which cannot be proven to be true from the axioms of arithmetic in a finite number of steps using a computerized theorem prover.

describing preferences and production technologies are restricted to be effectively computable and all choice sets, etc. are restricted to be recursively enumerable.[5]

A potential problem with Nachbar's and Lewis's results is that they require the computer to produce an *exact* solution to the problem under consideration. Most economic problems are formulated in terms of continuous mathematical entities (e.g. utility functions defined over orthants in Euclidean space), and it is generally impossible to compute exact solutions using a Turing machines or any other type of "universal computer" because they are fundamentally finite state devices which are only capable of finite precision arithmetic and can only perform a finite number of operations in any finite interval of time. However complexity problems arise even for completely discrete economic problems, i.e. where all variables are restricted to finite sets. These problems can be solved exactly on a Turing machine. For example Tsitsiklis and Athans (1985) considered the superficially simple

Team Decision Problem (TDP): Given finite signal spaces Y_1 and Y_2 and finite action spaces A_1 and A_2, and a discrete probabilty density $p(y_1, y_2)$, compute decision rules $\alpha_i : Y_i \to A_i$, $i = 1, 2$ that minimize the expected cost function

$$J(\alpha_1, \alpha_2) = \sum_{y_1 \in Y_1} \sum_{y_2 \in Y_2} c\left(y_1, y_2, \alpha_1(y_1), \alpha_2(y_2)\right) p(y_1, y_2). \tag{2.1}$$

Although it is not difficult to show that the TDP is effectively computable, Athans and Tsitsiklis proved that the TDP is is *NP-hard*, that is, it belongs to the same equivalence class of "difficult" computational problems such as the traveling salesman problem whose computational complexity is believed to grow exponentially fast in the size of the problem (in this case, the cardinalities of the sets (Y_1, Y_2, A_1, A_2)). Note that TDP is hard because of the restriction of complete informational decentralization: the two agents are not allowed to communicate their signals y_i which are treated as private information. However once we allow the agents to communicate their signals to each other or to a central planner, the TDP becomes "easy" in the sense that it can be solved in "polynomial time".[6] While Tsitsiklis and Athan's result certainly suggests the need for caution in claiming that decentralization is in some sense "computationally efficient," it appears that there are few realistic examples where the assumption of complete informational decentralization is plausible. While

[5] A recursively enumerable set is one for which there exists an effectively computable function which enumerates all the elements of the set.

[6] If a central planner observed the signal pair (y_1, y_2) she would simply choose the action pair (a_1, a_2) that minimized $c(y_1, y_2, a_1, a_2)$. This minimization problem requires at most $\|A_1\| \times \|A_2\|$ comparisons and function evaluations. The optimal decision rules (α_1, α_2) can then be computed as the functions giving the optimal action pair (a_1, a_2) for each possible state pair (y_1, y_2). Thus the total work involved in solving a centralized version of TDP is bounded by $(\|Y_1\| \times \|Y_2\| \times \|A_1\| \times \|A_2\|)$, which clearly grows polynomially in the size of the problem.

it is certainly true that agents in most economic agents almost never have complete information, they almost always have at least partial information obtained from messages from other economic agents. For example in models of competitive equilibrium, agents have access to a vector of prices, which can be viewed as "public broadcasts" of a set of signals which are functions of the collective information in the economy. The First Theorem of welfare economics tells us that in competitive equilibrium the price vector is a "sufficient statistic" for the collective private information in the economy in the sense that individual agents can make the right decision based only on their individual private information and the global information contained in the equilibrium price vector.

Unfortunately, results from the theory of continuous or *information-based* complexity show that even if agents were to truthfully report their private information to an "auctioneer" or "central planner", and even if we impose the less demanding requirement of computing an ϵ-approximation to a solution rather than the exact solution, the computational problems involved in computing a Walrasian equilibrium or a solution to a social planning problem can be shown to be *intractable*, in a sense to be precise below. In order to do this I briefly summarize a few key concepts from the theory of information-based complexity (IBC) below, referring the reader to the excellent monograph by Traub, Wasilkowski and Wózniakowski (TWW, 1988) for a complete account.

Even if we assume we had access to an idealized computer capable of storing arbitrary real numbers and performing infinite precision arithmetic, the solutions to most continuous mathematical problems such as integration, optimization, and zeros of nonlinear equations, fixed points of correspondences, etc. generally require information on what is essentially an infinite-dimensional quantity (i.e. the entire graph of the function f that is to be integrated, optimized, etc.). Since any physical computer system and realizable algorithm will be able to store and use information on f only at a finite number of points in its domain, (each of which are costly to acquire), we face a situation where the solutions to most continuous mathematical problems cannot be computed exactly but must be approximated. In addition, in many circumstances we don't have complete information about the function f, but only know that it belongs to some set of functions $\mathcal{F}$ such as the set $\mathcal{F} = C(X)$ of all continuous, uniformly bounded functions on some set X. If ϵ denotes a measure of the distance between the approximate and exact solution to the mathematical problem, then the worst case ϵ-complexity of a mathematical problem, $comp(\epsilon)$, is defined as the minimal cost of computing an ϵ-approximation to the true solution for any possible $f \in \mathcal{F}$.[7] The complexity function, $comp(\epsilon)$, represents the minimal computation time or *cost function* of producing

[7] Computer scientists have also studied the *average case complexity* of various mathematical problems in which a prior distribution is introduced over the set of possible functions $\mathcal{F}$ and the weaker requirement that the *expected approximation error* (where the expectation is taken with respect to the prior distribution) be less than ϵ.

an ϵ-approximation where the cost of calculations involved in producing an ϵ-approximation are defined in terms of a primitive model of the cpu time/space costs involved in the elementary operations needed to implement the solution algorithms. Many mathematical problems also have one or more associated parameters, represented here as d, denoting the dimension of the Euclidean space in which the mathematical problem lives. For example we could consider the problem of integration, optimization, or the problem of finding zeros to functions of d variables. In these cases it is customary to write the problem dimension d as an extra argument of the complexity function, e.g. $comp(\epsilon, d)$. Computer scientists say a computational problem is *intractable* if its complexity function takes the form

$$comp(\epsilon, d) = o\left((1/\epsilon)^d\right), \tag{2.1}$$

where the small o symbol denotes a lower bound on the complexity function. That is, an intractable problem is one for which the lower bound on the computation cost increases exponentially with the problem dimension d. In economics, problems that have this property are typically referred to as suffering from the *curse of dimensionality.*

Recent results in information based complexity theory (summarized in TWW, 1988) have established that a large number of mathematical problems are subject to the curse of dimensionality: multivariate integration, optimization, finding zeros of nonlinear functions, finding Brouwer and Kakutani fixed points, solution of partial differtial equations and Fredholm integral equations, and the problem finding fixed points to certain classes of contraction mappings (which includes the problem of solving infinite horizon continuous-state dynamic programming problems as a special case), all have worst case complexity functions of the form given in equation (2.1).[8]

It appears to be a relatively straightforward exercise to translate these general complexity bounds into corresponding complexity bounds for economic problems: utility maximization requires solution of constrained optimization problems, maximizing expected utility requires calculation of multivariate integrals, finding Walrasian equilibria requires calculation of zeros to an aggregate excess demand function or finding a Brouwer fixed point, calculation of rational expectations equilibria involve solutions to Fredholm integral equations, computation of option values involve solutions of PDE's, solutions to infinite horizon dynamic programming problems involve computation of contraction fixed points, and calculation of Nash equilibria reduce to the calculation

[8] See, e.g. TWW, 1988 for results on the intractability of multivariate integration, Nemirovsky and Yudin, 1984 on the intractability of constrained optimization, Sikorsky, 1985 on the intractability of zero finding, Chow and Tsitsiklis, 1992 on the intractability of dynamic programming, Hirsch and Vavasis (1989) on the intractability of Brouwer fixed points, Werschulz (1993) on the intractability of PDE's and Fredholm integral equations. See also Papadimitriou (1994) for related results on the intractability of Kakutani fixed points and the problem of finding approximate competitive equilibria in the discrete computational complexity framework.

of a fixed point to a correspondence. However to date formal complexity bounds have been established only for a relatively small number of economic problems such as social planning (Friedman and Oren, 1995), Walrasian equilibrium (Papadimitriou, 1995), and dynamic programming (Chow and Tsitsiklis, 1989). However it is likely that this list will quickly grow and formal proofs will soon be available showing that the majority of economic problems are intractable.

The next step is to translate complexity bounds into "impossibility theorems" stating that rational or optimizing behavior is impossible. However for reasons I discuss in section 3 below, it is generally not possible to prove a formal theorem to this effect, but one can make a strong argument that if rational, optimizing, or equilibrium behavior isn't impossible, then at the very least it is highly unlikely. The "impossibility argument" is roughly as follows. Real economic problems are extremely high dimensional. For example if d refers to the number of goods and services in the actual economy, then d is certainly as large as one hundred thousand and probably is closer to several hundred million depending on how narrowly we define a good or service. In many problems d will also be a function of the number of agents, which is well over 5 billion if we view the planet as a single integrated economy. To the extent that we want to compute a reasonably accurate approximation to the competitive equilibrium solution, (i.e. the level of approximation error tolerated, ϵ, is a small number such as $\epsilon = .1$), and for realistic values of d, the exponential complexity bounds, $o(1/\epsilon^d)$, imply that an astronomical number of calculations are required. The simple logic of exponential growth tells us that regardless of the values of the (typically unspecified) bounding constants in the complexity bounds, as d increases the number of calculations quickly grows so large that the world's fastest supercomputers would be unable to find an approximate solution to the problem in any reasonable period of time. To the extent that we believe that individual economic agents and the economy as a whole are subject to the same computational limitations, then these complexity bounds also constitute arguments for the impossibility of rational or equilibrium behavior. Simon's work on *satisficing* behavior is motivated by exactly this type of argument:

> "What a person *cannot* do he or she *will not* do, no matter how strong the urge to do it. In the face of real-world complexity, the business firm turns to procedures that find good enough answers to question whose best answers are unknowable. Because real-world optimization, with or without computers, is impossible, the real economic actor is in fact a satisficer, a person who accepts "good enough" alternatives, not because less is preferred to more but because there is no choice. Many economists, Milton Friedman being perhaps the most vocal, have argued that the gap between satisfactory and the best is of no great importance, hence the unrealism of the assumption that the actors optimize does not matter; others, including myself, believe that it does matter, and matters a great deal." (Simon, 1996, pp. 28–29).

It's difficult to dismiss impossibility arguments based on computational complexity bounds because we have observed the curse of dimensionality "empirically" in numerous numerical applications. The discovery that so many mathematical problems are computationally intractable provides a very compelling explanation why even the most gifted researchers in economics, physics, artificial intelligence and many other fields have been forced to limit their computational experiments to relatively small and simple "toy models" despite having access to what would seem to be very large computational resources (e.g. supercomputer centers, clusters of fast workstations, etc.). Furthermore the complexity bounds tell us that these problems are not just due to the fact that our computational methods are in an early state of development and new methods will ultimately be discovered which "crack" the problem: they constitute lower bounds on computation time using *any possible algorithm* regardless of whether or not these algorithms have been discovered. [9] In the next section I suggest some ways for dealing with complexity in the context of economic calculations.

3. Strategies for Dealing with the Complexity of Economic Calculations

In this section I argue that the worst case complexity bounds described in the previous section do not constitute proofs that rational or optimizing behavior impossible. I outline several different ways in which one can circumvent or break the curse of dimensionality underlying many economic calculations. Some of these methods may eventually make it possible to calculate approximate solutions to detailed, large scale economic models. Wherever possible I attempt to provide empirical evidence that the general strategies I outline have been successful in enabling economists and economic agents to solve hard problems.

It should be fairly obvious why the computational complexity bounds discussed at the end of section 2 do not constitute proofs that rational optimizing and equilibrium behavior is impossible: the question of whether or not the solution of a given economic problem is "computationally feasible" depends on a number of particulars that can only be determined on a case to case basis. These particulars include the processing speed of the available hardware, the amount of time available to perform the calculations, the level of error ϵ tolerated, the magnitudes of the bounding constants entering the complexity bounds, and whether the specific problem being solved represents the worst case, best case, or "average case". So it is ultimately an empirical question whether it possible to solve a specific economic problem given the available time, required accuracy, and available hardware. However this is a two-edged sword: if the proposition that "the intractability

[9] This is not quite true for the theory of discrete computational complexity where the famous $P = NP$ problem is still unresolved. If $P = NP$ then there is some "breakthrough" algorithm that enables us to solve all NP-complete problems (including the traveling salesman problem) in polynomial time. However to date no such algorithm has been discovered and most computer scientists believe no such algorithm exists, i.e. that $P \neq NP$. Unfortunately nobody has yet been able to prove the latter assertion either.

of economic problems implies that rational optimizing and equilibrium behavior is impossible" reduces to an empirical issue whose truth or falsity cannot be settled on purely logical grounds, then the proposition that "there are ways to break the intractability of economic problems and this implies that rational optimizing and equilibrium is possible" is also an empirical question that cannot be settled by deductive reasoning. The best and most convincing "proof" of the latter proposition is to show that there exist hardware and software that can actually solve realistically specified optimization and equilibrium problems. However I want to emphasize the point that the fact that rational behavior is possible does not imply that all economic agents are actually behaving approximately optimally or all markets are approximately in equilibrium: this is another empirical question that is beyond the scope of this essay. My objective in this section is much more limited: I argue that the predictions of neoclassical economic theories are not "impossible" by suggesting a number of ways of dealing with the complexity of economic calculations.

3.1 "Hardware Solutions" to the Problem of Economic Complexity

The "hardware solution" is based on the following two facts: a) along most dimensions human brains are orders of magnitude more powerful than the fastest man-made computers, and b) the power of man-made computers is growing at an exponential rate. Fact a) opens up the possibility that even though it is impossible to solve realistic versions of many economic problems using current computers in a reasonable amount of time, the human brain may be sufficiently powerful to solve these problems in the available time. Fact b) suggests that man-made computers will eventually catch up and exceed the power of the human brain, so that solving realistic versions of economic problems may indeed eventually be possible even if various economic optimization and decision problems are subject to a curse of dimensionality. Evidence for fact a) comes from research in neuroscience which shows despite the fact that switching times and propagation speed of electrical impulses in neurons are orders of magnitude slower than corresponding electrical signals in silicon chips (which travel at nearly the speed of light), the brain is nevertheless an amazingly powerful "chemical computer" whose overall information processing capabilities are orders of magnitude faster than the fastest man-made computers constructed from silicon microprocessors:

> "Computation is also limited by power consumption, and on this matter too the brain is impressively efficient. For example, a neuron uses roughly $10 \times e - 15$ joules of energy per operation (e.g. one neuron activating another at a synapse). By contrast, the most efficient silicon technology currently requires about $10 \times e - 7$ joules per operation (multiply, add, etc.) (Mead, 1989). Using the criterion of joules per operation the is about *7 or 8 orders of magnitude* more power efficient that the best of the silicon chips. A direct consequence of their energy efficiency is that brains can perform many more operations per second than even the newest supercomputers. The fastest digital computers are capable of of around $10 \times e + 9$ operations per second; the brain of the common housefly, for example, performs about $10 \times e + 11$ operations per second when merely resting." (Churchland and Sejnowski, 1992, p. 9).

Fact b) is primarily a result of advances in photolithography which have enabled computer manufacturers to ultraminiaturize integrated circuits, leading to exponential growth in the speed and memory of digital computers: "Microprocessor performance has increased by 4X every three years, matching the rate of integrated circuit logic density as predicted by Moore's law. For example the microprocessors of 1993 are around 200 times faster than those of 1981." (National Science Foundation, 1993, p. 25). Although it is generally recognized that we are starting to reach a point of diminishing returns in the amount of computational power that can be built into a single cpu, which is partly due to phyiscal limits to the degree of miniaturization due to optical lithography, and partly due to limits on how short a cpu clock cycle can be, it is hazardous to predict a permanent slowdown in the growth rate of cpu power given that there are no known physical lower bounds on the amount of energy required for computation, and due to new technologies such as optical, and quantum-based computing devices, and even self-assembling molecular scale "organic" computers that could lead to circuit densities that are 10,000 time greater than current microchips. [10] The recent victory of IBM's "Deep Blue" computer over the top ranked human chess player Gary Kasparov appears to foreshadow the day when man-made computers will be able to outperform the human brain in terms of overall information processing capacity.

3.2 "Software Solutions" to the Problem of Economic Complexity

Obviously the mere fact that we have access to very powerful hardware doesn't necessarily imply that we have the capacity to solve hard problems. The possibility of rational behavior becomes more compelling if we can show that there are ways of circumventing or breaking the curse of dimensionality underlying many economic calculations. In this section I outline a number of "software solutions" that succeed in making otherwise intractable problems tractable. Specifically, a problem is said to be *tractable* if its complexity $comp(\epsilon, d)$ is bounded above by a polynomial in the problem dimension d and the desired accuracy level ϵ. Due to the logic of exponential growth, it will generally take orders of magnitude less work to solve a high-dimensional problem whose complexity increases polynomially fast in d than a problem whose complexity increases exponentially fast in d. Below I show that for alternative measures of complexity, problems which appeared to be intractable under one definition of complexity can become tractable under an alternative formulation. The use of different complexity measures is not simply a clever trick designed to define our way out of the problem: rather, different complexity measures correspond

[10] See, e.g. Reed *et. al.* (1997). If development of self-assembling molecular-based computing devices proves economically viable, it likely that we will highly "brain-like" computer hardware that involves a truly massively parallel architecture. However controlling and programming these systems will present some very formidable software challenges, and this is where the use of "decentralization" may prove to be essential as will be discussed in section 4.

to different ways of assessing the "quality" of an approximate solution (i.e. do we demand that an algorithm produces an ϵ-approximation with probability 1, or are we content to use a method that finds an ϵ-approximation with sufficiently high probability?), as well as different amounts of prior information about the problem and its solution. In particular, I will show that the amount of prior information at our disposal can have a very big impact on the types of methods we use to solve a problem, so whether the "software solutions" outlined below actually imply that rational behavior is possible becomes an empirical question about the amount of prior information we possess about a problem.

3.2.1. Exploitation of special structure. Most of the known lower bounds on the worst case complexity of continuous mathematical problems have been derived in the context of information based complexity theory using an "adversary principle". In rough terms, the adversary can be viewed as an "opponent" who selects the worst case problem from a set $\mathcal{F}$ of possible problems in response to any algorithm and level of computational effort chosen by and level of computational effort. The complexity of a problem is then the minimax solution to the "game" between the adversary and the person performing the computations. The complexity of the problem is typically very sensitive to the size of the set $\mathcal{F}$ of possible problems: the more latitude we give the adversary in choosing a worst case scenario the higher the overall complexity of the problem. Conversely, by restricting $\mathcal{F}$ to a smaller set of problems which has some time of "special structure" we can often reduce the complexity of the problem, sometimes very dramatically. For example consider the consumer's problem:

$$max \left\{ u(x) | p'x \le y \right\}, \quad x \in R^d_+ \tag{3.1}$$

An ϵ-approximation to this global optimization problem is a vector $\hat{x}$ satisfying $p'\hat{\le}y$ and $|u(\hat{x}) - u(x^*)| < \epsilon$, where x^* is an vector which attains a global maximum in equation (3.1). The following result is from Nemirovsky and Yudin (1983):

Theorem 3.1: *Let $\mathcal{F}$ denote the set of r-times continuously differentiable functions on R^d_+. Then a lower bound on the worst-case deterministic complexity of the consumer's problem (3.1) is given by:*

$$comp(\epsilon, d, r) = o\left((1/\epsilon)^{d/r}\right) \tag{3.2}$$

If $\mathcal{F}$ is the set of all concave functions on R^d_+ then the worst-case deterministic complexity of the consumer's problem is given by:

$$comp(\epsilon, d, r) = \Theta\left(d \log(1/\epsilon)\right). \tag{3.3}$$

where Θ denotes both an upper and lower bound on complexity.

In other words the consumer's problem is intractable in the general case when we allow any possible smooth utility function, but the problem becomes tractable when we exploit the fact that utility functions have special structure like concavity or monotonicity.[11] In fact, the optimization problem in the latter case is a fairly easy problem since complexity is not only polynomial but linear in the number of commodities d.

Whether or not it is reasonable to believe that utility functions are always concave is an empirical question that I don't attempt to address here. The key point of this example is that it calls into question the relevance of worst case complexity bounds derived for very general classes of problems $\mathcal{F}$. In practical situations the worst case bounds will may be far too pessimistic, since it ignores the possibility that economic agents will exploit prior knowledge about the special structure of a problem, enabling them to solve it far more rapidly than predicted by the worst case bounds. Worst case bounds do not seem to be relevant in explaining observed performance of algorithms in number of different problems. For example Klee and Minty (1972) constructed a subfamily of linear programming problems which essentially forced the simplex algorithm to visit nearly all vertices of the constraint set before it could find the optimal solution. Since the number of vertices increases exponentially fast as the number of constraints and variables in an LP increases, their counter example shows there are worst case problems for which the simplex algorithm is computationally intractable. Even though there are other algorithms (such as interior point methods) which are known to have polynomial time performance in the worst case, experience has shown that in most practical problems the simplex algorithm is much more efficient than other methods which are supposed to guarantee and works remarkably well, visiting only a very small subset of vertices before locating the optimal solution. The worst case results seems to yield misleading conclusions about the difficulty of linear programming and the efficacy of the simplex algorithm, especially if we believe that it is good at exploiting the special structure of problems we typically encounter in practice.

Another example is high dimensional numerical integration. As discussed in section 2, it is well known that multivariate integration using deterministic methods has a worst case complexity that grows exponentially fast in the problem dimension, d. These bounds and practical experience have lead to the conventional wisdom (see, e.g. Davis and Rabinowitz, 1977) that deterministic integration methods work well only in fairly low-dimensional problems, such as for $d \leq 5$. High dimensional integration problems arise regularly in economics, such as in computing expected discounted values of random payouts that are required to price various financial instruments. In a

[11] Nemirovsky and Yudin's result did not consider the additional gains to exploiting monotonicity of the utility function.

recent paper Paskov and Traub (1995) used a deterministic integration method to compute expected present values of collateralized mortgage obligations (CMOs) which yield monthly cash flows over 30 year horizons, amounting to integration in $d = 360$ dimensions. One of the integration methods they used was a simple average of the integrand evaluated at N deterministically chosen points known as the *Sobol points*. They found that they could get accurate approximations of this 360 dimensional integration problem for N as small as 50,000. This example suggests that even though all deterministic integration methods are provably intractable on a worst case basis, the worst case complexity bounds seem to be irrelevant in the CMO case. That is, we don't observe an exponentially fast increase in the solution times as the problem dimension d increases, which suggests that the behavior of this algorithm is better approximated by an "average case" than worst case analysis.[12]

3.2.2. Decomposition. This approach is closely related to the strategy of exploiting special structure. The idea is certainly not new: similar ideas can be found in the work of Simon and Ando (1961) on *nearly decomposable systems*. Decomposition is a method that is motivated by the fact that it is generally far easier to solve a large problem by breaking it up into a union of approximately independent subproblems rather than attempt to solve the overall problem directly. The clearest way to illustrate the potential gains from adopting this strategy is to consider an example exhibiting perfect decomposability. Consider a dynamic programming problem with the following utility function, transition probability, and choice sets:

$$u(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{N} u_i(s_i, a_i) \tag{3.4}$$

$$p(\mathbf{s'}|\mathbf{s}, \mathbf{a}) = \prod_{i=1}^{N} p_i(s_i'|s_i, a_i) \tag{3.5}$$

$$A(\mathbf{s}) = A_1(s_1) \times A_2(s_2) \cdots \times A_N(s_N). \tag{3.6}$$

In this example we can think of the overall problem as consisting of N tasks. The vector s_i represents the state of "task" i and the boldface **s** and **a** denote the vectors of states and actions for all N tasks, $\mathbf{s} = (s_1, \ldots, s_N)$ and $\mathbf{a} = (a_1, \ldots, a_N)$. Equation (3.5) states that task states evolve independently. A practical example of such a problem might be a large rental car company

[12] One can prove that multivariate integration is tractable on an average case basis, where particular (typically Gaussian) priors are used to represent the likelihood of encountering various types of integrands (see, e.g. TWW, 1988 or Paskov, 1993).

which must determine an optimal replacement policy for each vehicle in its fleet. Intuitively the separability and independence assumptions imply that the problem is perfectly decomposable into the N separate subtasks, each of which can be solved independently to construct the overall solution. It is straightforward to verify this formally. The *value function* for the dynamic programming problem is given by:

$$V(\mathbf{s}) = \max_{\mathbf{a} \in A(\mathbf{s})} \left[u(\mathbf{s}, \mathbf{a}) + \beta \int_{\mathbf{s'}} V(\mathbf{s'}) p(\mathbf{ds'}|\mathbf{s}, \mathbf{a}) \right]. \tag{3.7}$$

Lemma 3.1: If $u(\mathbf{s}, \mathbf{a})$ has the additive decomposition given in (3.4) and $p(\mathbf{s'}|\mathbf{s}, \mathbf{a})$ satisfies the multiplicative decomposition given in (3.5), and the constraints sets $A(\mathbf{s})$ have the product structure given in (3.6), then the value function $V(\mathbf{s})$ has an additive decomposition given by:

$$V(\mathbf{s}) = \sum_{i=1}^{N} V_i(s_i) \tag{3.8}$$

where each V_i is given by:

$$V_i(s_i) = \max_{a_i \in A_i(s_i)} \left[u_i(s_i, a_i) + \beta \int_{s_i'} V_i(s_i') p_i(ds_i'|s_i, a_i) \right]. \tag{3.9}$$

Lemma 3.2: *The worst case complexity of a continuous-state decomposable dynamic programming problem satisfying* $(3.4), \ldots, (3.6)$ *is given by:*

$$comp(\epsilon, d_a, d_s, N) = N\Theta\left((1/\epsilon)^{d_a + 2d_s}\right) \tag{3.10}$$

where d_a *denotes the dimension of* a_i *and* d_s *denotes the dimension of* s_i *(all assumed to have the same dimension for simplicity). The worst case complexity of a non-decomposable dynamic programming problem that doesn't have the product structure in* $(3.4) \ldots (3.6)$ *is given by:*

$$comp(\epsilon, d_a, d_s, N) = \Theta\left((1/\epsilon)^{N(d_a + 2d_s)}\right). \tag{3.11}$$

Lemma 3.1 confirms the intuition that perfect decomposability allows us to independently solve each subproblem and then combine the solutions to the subproblems to obtain a solution for the overall problem. Lemma 3.2 quantifies the gains from doing this. Whenever the overall problem is subject to the curse of dimensionality it is much easier to solve the N separate subproblems with dimensions $(d_1, \ldots, d_N)$ than it is to solve one large problem with dimension $d_1 + \cdots + d_N$: complexity grows linearly in the number of tasks N when the problem is solved via decomposition but exponentially in N otherwise.

The principle of decomposition is closely associated with parallel processing, since the subproblems can obviously be solved on separate processors. This technique is widely used in physical and engineering applications where it is known as "domain decomposition" since the method is commonly used to solve PDE's and problems which have a spatial structure which allows one to identify subproblems based on partitions of the domain.[13]

Of course most problems cannot be exactly decomposed into completely independent subproblems. The challenge is to recognize whether or not a problem is nearly decomposable, and if so, to identify its approximately independent subproblems, determine whether they can be solved separately (ignoring any possible interdependencies between the subproblems), or quasi-separately (summarizing feedbacks from the other subproblems using relatively low dimensional sets sufficient statistics). Simon (1996) notes that "in the natural world nearly decomposable systems are far from rare" and in economics, there are many cases where it is easy to identify sufficient statistics which summarize the impact of other subsystems in the economy:

> "It is empirically true that the price of any given commodity and the rate at which it is exchanged depend to a significant extent only on the prices and quantities of a few other commodities, together with a few other aggregate magnitudes, like the average price level or some over-al measure of economic activity. The large linkage coefficients are associated in general with the main flow of raw materials and semifinished products with and between industries. An input-output matrix of the economy, giving the magnitudes of these flows, revals the nearly decomposable structure of the system — with one qualification. There is a consumption subsystem of the economy that is linked strongly to variables in most of the other subsystems. Hence we have to modify our notions of decomposability slightly to accomodate the speical role of the consumption subsystem in our analysis of the dynamic behavior of the economy. (Simon, 1996, pp. 28–29).

Introspection suggests that our brains make extensive use of decomposition, with specialized areas of the brain which are highly optimized for solving specific subproblems such as visual and auditory processing, speech, coordination, and so forth. Although it is not easy to identify physical brain structures associated with "higher level" mental processing in the cerebral cortex, it still appears that our brain does a very effective job of solving many different subproblems in parallel and assembling the solutions to these subproblems to yield a coherent pattern of overall behavior. For example in the course of a day we do quite a number of different activities and it might be reasonable to treat these as approximately independent subproblems. For example, the problem of deciding which articles to read in the morning newspaper seems independent of the problem of which route I should take to work. One can imagine interdependencies that might arise, say, if the newspaper contained an article on construction delays on the expressway, but ordinarily it may be a good approximation to assume that interlinkages don"t exist if the cost to ignoring the interlinkages is small. It seems quite unlikely that we solve a single massive dynamic programming problem *ex*

---

[13] For further information on this literature, see the web site at `www.ddm.org`.

ante that provides us with an overall contingency plan for dealing with any possible eventuality. Instead, it seems that we must be doing something closer to solving a fairly large number of subproblems corresponding to different activities we engage in, and accounting for unexpected interlinkages between problems by recomputing the solution to a particular subproblem “on the fly” whenever new information arrives that suggests that our previous solution to the subproblem may not be a good one.

I believe that there could be very high payoffs to research that would help us understand how our brain is able to decompose large, complex problems into a number of tractable subproblems and then “reassemble” the individual solutions into a sensible solution to the overall problem. It seems that economists could particularly benefit from a better understanding of how the brain succeeds in piecing together solutions to subproblems, since it might help them to discover ways to integrate the insights gained from numerous “micro models” that provide relatively detailed models of small aspects of economic behavior with the insights of highly aggregative “macro models” to obtain much more detailed yet coherent overall models of macroeconomic activity. In my opinion economists have been unsuccessful in modularizing and interconnecting specialized models of subcomponents of the economy into a sensible model of the overall economy, and this failure is one of the key factors limiting the rate of advancement and practical usefulness of economic research. In the conclusion I suggest that the adoption of a high level protocol that provides a standard interface for economic models to intercommunicate could go a long way towards making economic science a more cumulative collective endeavor, reducing the need to continually to “reinvent the wheel” by allowing new researchers to more easily integrate and build upon results of previous researchers.

3.2.3. Randomization. There are certain problems which are known to be intractable in the worst case using deterministic algorithms but which can proven to be tractable when randomized algorithms are employed. The classic example of a randomized algorithm is monte carlo integration. The central limit theorem implies that the monte carlo integral converges to the true integral at rate $1/\sqrt{N}$ (where N is the number of sample points used in the monte carlo integral), regardless of the the number of variables in the integrand, d. The drawback of randomized algorithms is that they are not capable of generating an ϵ-approximation with probability 1, but instead we must settle for the weaker assurance that the expected error of the algorithm is less than ϵ.[14] However in some cases there are substantial payoffs for settling for this some what weaker guarantee as summarized in the following result (for details see TWW 1988):

[14] By appealing to Markov’s inequality, it is easy to show that a randomized algorithm is capable of generating an ϵ-approximation with probability $1-\delta$, where ϵ and δ are arbitrarily small solution tolerances.

Theorem 3.2: *Consider the problem of multivariate integration of the class $\mathcal{F}$ of r-times continuously differentiable functions on the d-dimensional hypercube, $[0,1]^d$ with uniformly bounded $r+1^{st}$ derivative. Then the worst case deterministic complexity is given by:*

$$comp^{\textit{wor-det}}(\epsilon, d, r) = \Theta\left(1/\epsilon^{d/r}\right) \tag{3.12}$$

and the worst case randomized complexity is given by:

$$comp^{\textit{wor-ran}}(\epsilon, d, r) = \Theta\left(\frac{\gamma(d)}{\epsilon^m}\right), \tag{3.13}$$

where m is given by:

$$m = \frac{2}{\left(\frac{2r}{d}\right) + 1}.$$

Thus, if we restrict ourselves to deterministic algorithms such as quadrature the multivariate integration problem is intractable since complexity increases exponentially fast in d whereas with randomization the problem is tractable since the complexity is $O(1/\epsilon^2)$ independent of the dimension d.

The curse of dimensionality of deterministic methods such as numerical quadrature have been observed in practice. Standard texts on numerical integration (e.g. Davis and Rabinowitz, 1977) conclude that deterministic integration methods work well only in fairly low-dimensional problems, such as for $d \leq 5$ and recommend monte carlo integration for higher dimensional problems. High dimensional integration problems arise regularly in economics, such as in computing expected utilities in portfolio optimization problems or expected discounted values of uncertain monthly dividends coupons that are required to price various long-term financial instruments such as in the CMO pricing example discussed in section 3.2.2. [15] However it is important to note that randomization does not succeed in breaking the intractability of all types of mathematical problems. For example the randomized complexity of optimization problems is the same as the deterministic complexity: the problem is intractable in the worst case in both settings. An important open question is whether optimization is tractable in an average case setting. The results in section 3.2.1 suggest that exploiting prior information about special structure of an optimization problem can dramatically reduce its complexity. However it is difficult to specify priors that embody our intuitive notions about special structure such as monotonicity or unimodality in an analytically

[15] Although the deterministic, low discrepancy method that Paskov and Traub used to solve the CMO problem did not appear to obey the worst case complexity bounds and substantially outperformed monte carlo integrals (in apparent contradiction of Theorem 3.2), the reader should remember that the complexity bounds in Theorem 3.1 are for the worst case scenario and may not be relevant for understanding performance in particular cases which might represent average or even best case scenarios.

tractable manner. So far, most research into the value of prior knowledge about optimization problems has been done in a worst case framework and reflecting prior knowledge by restricting the the class of admissible functions $\mathcal{F}$.

Monte carlo integration can be extended to infinite-dimensional integration problems such as path integration (e.g. computation of the expected value of a functional of a Wiener process), and similar logic can be applied to show that randomization breaks the curse of dimensionality for these problems (see Wasilkowsky and Woźniakowski, 1996). Recently, randomization has been shown to break the curse of dimensionality involved in solving to functional equations arising from rational expectations models and dynamic programming problems. In the case of dynamic programming problems and any type of problem involving conditional expectations that depend on state variables that can assume a continuum of values there are essentially infinitely many integrals that need to be approximated, e.g. on integral for each possible value of s in order to compute a value function $V(s)$. However Rust (1997a,b) showed that the logic of monte carlo integration extends naturally to dynamic programming problems: we approximate V, the unique fixed point to the *Bellman operator* $\Gamma(V)$ given by:

$$\Gamma\left(V\right)(s) = \max_{a \in A(s)} \left[u(s,a) + \beta \int V(s') p(s'|s,a) ds' \right] \tag{3.14}$$

by $\hat{V}_N$ an approximate fixed point to the *random Bellman operator* $\hat{\Gamma}_N(V)$ given by:

$$\hat{\Gamma}_N\left(V\right)(s) = \max_{a \in A(s)} \left[u(s,a) + \frac{\beta}{N} \sum_{i=1}^{N} V(\tilde{s}_i) p(\tilde{s}_i|s,a) \right] \tag{3.15}$$

where $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$ are *IID* uniform draws from the d-dimensional state space $S = [0,1]^d$. Note that although $\hat{\Gamma}_N(V)$ is a random function of s, we can still apply a *functional central limit theorem* to show that $\|\hat{\Gamma}_N(V) - \Gamma(V)\| = O_p(1/\sqrt{N})$ independent of the dimension d of the state space S. This result, the triangle inequality, and the contraction mapping property imply that $\|V - \hat{V}_N\| = O_p(1/\sqrt{N})$, where $\hat{V}_N$ is the fixed point to $\hat{\Gamma}_N$. This in turn implies the following result:

Theorem 3.3: *If the constraint sets $A(s)$ are finite with a uniformly bounded number of elements for all $s \in S$, then under certain Lipschitz regularity conditions on u and p we have:*

$$\begin{aligned} comp^{\textit{wor-det}}(\epsilon, d) &= o\left(1/((1-\beta)^2 \epsilon^{2d}\right) \\ comp^{\textit{wor-ran}}(\epsilon, d) &= O\left(\frac{d^4}{(1-\beta)^8 \epsilon^4}\right). \end{aligned} \tag{3.16}$$

So at least for dynamic programming problems with finite numbers of possible actions, randomization succeeds in breaking the curse of dimensionality since the worst-case deterministic complexity increases exponentially fast in d but the worst-case randomized complexity increases polynomially in d. The theoretical predictions in Theorem 3.3 have been verified in numerical experiments in Rust (1997b), which showed that randomized solutions to the problem outperformed standard deterministic solution methods even in relatively low-dimensional test problems with $d = 2$. [16] This is an encouraging finding since for ordinary integration monte carlo methods are generally recommended only for problems with $d > 5$ since deterministic methods are faster and more accurate for low dimensional cases. However Rust's results show that the performance of randomization is critically dependent on the degree of smoothness of the transition density p and that in problems where the Lipschitz bound for p increases exponentially fast in d the curse of dimensionality reappears even when randomization is used.

Another advantage of the approximation strategy in equation (3.15) is that the method dramatically reduces the memory requirements necessary to obtain a uniform approximation to the true solution V. The approximate value function $\hat{V}_N$ can be summarized by only N numbers, $\{\hat{V}_N(\tilde{s}_1), \ldots, \hat{V}_N(\tilde{s}_N)\}$, the values of $\hat{V}_N$ on the set of random grid points $\{\tilde{s}_1, \ldots, \tilde{s}_N\}$. Once these are computed the value of $\hat{V}_N$ and the corresponding approximate decision $\hat{\alpha}_N$, can be computed "on the fly" for any alternative point $s \in S$ in only $O(N)$ operations using (3.15). To calculate an approximate value of V at an arbitrary $s \in S$ using standard approaches to discretization of dynamic programming problems would involve interpolation from the approximate value function calculated on grid containing $o(1/\epsilon^d)$ points. Thus, randomization allows us to break a curse of dimensionality involved in memory requirements and cpu consumption simultaneously. This is important since memory constraints may be just as important as time constraints motivating the use of more effective algorithms in digital computers and the human brain. This example shows that there exist economical ways of storing information that will allow an agent with bounded memory capacity and processing speeds to take nearly optimal actions in states they have never encountered before: there is no need for agents to store elaborate "contingency tables" to behave nearly optimally.

The use of randomization to solve high dimensional and large scale problems has long history in physics and chemistry (beginning with the pioneering work by Ulam, Metroplis and von Neumann in Los Alamos in the 1940's), but economists have only begun to adopt monte carlo

[16] However certain deterministic methods based on Sobol points outperformed the randomized method. This is another example indicating that worst case deterministic complexity bounds may not be relevant for understanding the performance of algorithms in specific cases. The problem-specific convergence bounds derived in Rust (1997b) provided a much more accurate prediction of the actual behavior of various algorithms than the worst case bounds.

techniques and simulation methods comparatively recently. Already these methods have had a large impact in econometrics, where "simulation estimators" have significantly extended the size and complexity of models that can be estimated. Stochastic learning algorithms such as Q-learning, temporal difference learning and "real time dynamic programming" are playing an increasingly important role in the artificial intelligence literature. Although there are no formal proofs that these stochastic algorithms succeed in breaking the curse of dimensionality, numerical experiments using similar methods to solve a class of dynamic games in Pakes and McGuire (1996) suggest that this might be the case. A problem is has table lookups for states which are encountered, integrating it with random Bellman approach might be effective.

It is difficult to determine the extent to which nature employs randomization is as a tool for solving hard problems. Randomization seems to play a key role in evolution, with genetic mutations serving as a driving force for experimentation and improvement. Brain signals appear to have a large random component, "Most neurons are spontaneously active, spiking in random intervals in the absence of input" (Churchland and Senjowski, 1992, p. 53). Furthermore the neuron cell body is known to cumulate or add the random stimulative and inhibitory signals received from other neurons via the thousands of synapses on its dendrites until a certain potential threshold is achieved that causes the neuron to fire. This at least suggests the possibility that some form of "monte carlo" integration may be playing a role in the operation of the brain. Randomization also plays a key role in current theories of asssociative memory and problem solving. These theories model neural networks in the brain as stochastic "Boltzmann machines" or "Hopfield networks" which are essentially nonlinear dynamic systems that implement versions of simulated annealing in search of stable, energy minimizing configurations of the network. These energy minimizing configurations are thought to represent memories, solutions to problems, etc. Random shocks play a key role in simulated annealing, preventing the dynamic system from converging to local optima of the "energy landscape".

Although we noted above that randomization does not break the curse of dimensionality of optimization, in my "soap film" example in section 4 I suggest that thermodynamic and electo-dynamic systems can also be viewed as massively parallel processors which succeed in solving highly complex global optimization and equilibrium problems extremely rapidly. To the extent that random noise plays an important role in the dynamics of these systems and their ability to find equilibrium or energy minimizing configurations, then there may be a good reason why we seem to observe a prominent role played by randomization in natural systems.

3.2.4. Knowledge Capital. There are very few problems that we have to solve completely "from scratch." We often have access to a solution to a similar problem that provides a good initial guess or starting point to help us solve the problem at hand. In some cases previously discovered solutions are so good that we don't have to expend any computational effort at all: we simply "copy" the previously discovered solution without even thinking about it. Stated differently, many potentially difficult decisions and economic calculations that we encounter in daily life are avoided because economic agents have access to substantial amounts of *knowledge capital.* The stock of knowledge capital can be thought of as a huge library of previously discovered solutions to various problems. By using the existing stock of knowledge capital we can avoid expending substantial amounts of computational effort "reinventing the wheel." Of course, the library rarely contains a solution to a problem exactly like the one we are confronting in any given situation, but provided the problem at hand is not too different from problems that have been previously encountered and "solved", it is often the case that by making relatively small modifications to these previously discovered solutions we can obtain a very good approximation to the solution of the problem at hand and, as I show below, save a lot of computational effort in the process.

There are many different examples and aspects of this relatively simple and intuitive point, but before discussing them I would like to present a simple example that formalizes the potential gains to using knowledge capital. Consider a dynamic version of an exchange economy with a sequence of static "temporary equilibria" which change from day to day as a result of stochastic shocks to preferences and endowments. Let $x_{it}(p)$ be the demand function of consumer i in period t and let e_{it} be the corresponding endowment. Then the equilibrium price vector at day t satisfies:

$$\sum_{i=1}^{N} x_{it}(p_t) = \sum_{i=1}^{N} e_{it} \tag{3.17}$$

If we try to compute the equilibrium price vector p_t from "scratch", such as in cases where we have no knowledge of previous temporary equilibrium prices for "nearby problems" (i.e. problems at other times $s < t$ for which $\|x_{it}(p) - x_{is}(p)\|$ and $\|e_{it} - e_{is}\|$ are small uniformly in p and i), then a by applying the worst case complexity bounds in Sikorski (1985) or Hirsch and Vivasis (1989) we conclude that the worst case complexity of the temporary equilibrium problem is $o(1/\epsilon^d)$ where d is the number of commodities. [17] Now consider a case of a market that has been in operation for a long time in a stationary environment so that by some adjustment process the market has been able to find an ϵ-approximation to the equilibrium price at $t-1$. If we also assume that a) the equilibria at days t and $t-1$ are regular (i.e. each consumer has a unique utility-maximizing bundle and the gradient of the excess demand function is non-singular at the equilibrium price

[17] Here we assume that the consumers' optimization problems are solved exactly and at zero cost in order to simplify the argument. The complexity of the problem would only increase if we also accounted for the cost of solving the consumers' optimization problems.

vector), b) the shocks affecting preferences and endowments are sufficiently small so that p_{t-1} is in a "domain of attraction" of the equilibrium price at day t, then the worst case complexity of finding an ϵ-approximation to the equilibrium price vector at day t increases polynomially rather than exponentially fast in d. For example one polynomial-time algorithm for computing p_t is Newton's method:

$$p_k = p_{k-1} - [E_t'(p_{k-1}]^{-1} E_t(p_{k-1}), \tag{3.18}$$

where p_k is the approximate solution at iteration k of Newton's methods and $E_t(p_{k-1})$ and $E_t'(p_{k-1})$ denote the value and gradient of the excess demand function at day t evaluated at the trial price vector p_{k-1}. Traub and Woźniakowski (1979) showed that provided the starting price vector $p_0 = p_{t-1}$ for Newton's method in the domain of attraction for the zero $E_t(p_t) = 0$, then an upper bound on the complexity of the Newton iteration is $O\left(\log\log(1/\epsilon)d^3\right)$, which is polynomial rather than exponential in d. Of course, I do not claim that decentralized price adjustment actually behaves according to Newton's method: my point is simply to illustrate there exist price adjustment procedures that can circumvent the curse of dimensionality when shocks to the market are sufficiently small. The market mechanism doesn't have to start from scratch in the process of computing an approximate equilibrium price p_t at each time t: instead, the previous price p_{t-1} ordinarily constitutes as very good initial guess for p_t.

There are even more compelling examples of how knowledge capital can vastly reduce the amount of computation required to operate sensibly in a complex environment. The most basic example is biological evolution, where DNA is probably the most obvious form of knowledge capital. DNA constitutes the "blueprint" that nature has "discovered" for constructing successful living organisms via a long run process of ecological competition and random trial and error (e.g. chance outcomes from cross-over and genetic mutations in sexual reproduction, as well as effects of stochastic changes in the environment). In addition to serving as a blueprint for the construction of organisms, DNA also "hardwires" many of an organism's behavioral patterns in the form of basic instincts or proclivities. As a result DNA limits the number of things an organism needs to "think about" and thereby helps an organism solve the potentially intractable problem of choosing a particular behavioral rule from the huge space of all possible behavioral rules in order to maximize its fitness. Examples of hardwired instincts include the sex drive, the instinct to "nest" and nurture young, the instinct of children to imitate their parents, the apparent "automatic" development of language skills in humans, and so forth. To the extent that hardwired behavior patterns are simply stored and do not need to be learned or recomputed, they can significantly reduce the organism's computational burdens. More importantly, hardwired behavior patterns limit the space of possible decision rules (alternative actions) that the organism must "consider" and this could vastly reduce the number of new computations that the organism must perform in order to take a particular course of action in real time. Most of the real-time computations that organisms perform seem to be restricted to a much a more limited set of actions that they need to take on a day by day and second by second to survive. The success of evolutionary processes suggests that the environment is sufficiently stationary that the knowledge capital embodied in DNA usually provides a suficiently good "starting value" to reduce the complexities faced by each succeeding generation of organisms in surviving and adapting to their environments.[18]

The DNA example also suggests reasons to doubt the relevance of impossibility results based on worst case analyses. I argue that evolutionary processes lead to organisms that have the capability to solve average case but not necessarily worst case problems. Even though there may exist worst case problems that organisms are unable to solve in the limited real time they have available (e.g. a gazelle isolated from the herd may not be able to think fast enough to outwit a cheetah hunting for its next meal), what matters for the overall fitness and survival of the species is that organisms be able to solve the average case problems they typically encounter (e.g. most gazelles will typically survive being hunted by cheetahs provided they are smart enough to run toward the middle of the herd). This suggests that in thinking about the impact of DNA and other forms of knowledge capital in an evolutionary context, the relevant question is whether the problems organisms encounter are tractable in an average case rather than worst case setting. If a species of organisms were unable to quickly find good solutions to "fitness-relevant" problems it typically encounters, then evolutionary processes would eventually drive the species into extinction, possibly by supplanting it with a new species of organisms that are able to solve these problems. In either case evolutionary processes lead to a form of "selectivity bias" where species which emerge as successful long-term survivors in a stationary environment must be capable of solving the fitness-relevant computational problems they typically confront. This suggests that the problems organisms solve must be tractable, at least in an appropriate average case setting.

In human economies and societies a large fraction of knowledge capital is fairly tangible and obvious: it includes the stock of patents and inventions that have been accumulated throughout history. Each new generation inherits the stock of ideas and inventions from the previous generation, enabling it to concentrate on the much more limited problem of developing new ideas and inventions that solve new or existing unsolved problems, thereby further adding to the stock of knowledge capital. There is also a less obvious, but probably equally important form of knowledge capital embodied in economic institutions such as the stock exchange and commodity markets,

[18] The dinosaurs are a prominent counterexample.

legal institutions and concepts such as limited liability and private property rights, government institutions such as Social Security, social institutions such as the family, and customs such as taboos on incest, adultery, and so forth. These institutions can be viewed as constituting approximate solutions to various problems that society as a whole is facing. Similar to DNA, these institutions have survived an evolutionary process of trial and error and competition. Some institutions such as social insurance appear to be in a relatively early stage of evolution, and seem to be undergoing frequent changes to enable them to provide better solutions to the socio-economic problems that motivated them. As a result the previous configuration of these institutions might not always necessarily represent a good starting point for solving today's problems, especially when society and the economy is changing rapidly.[19]

However there are other institutions such as the "market institution" that appear to have converged to relatively stable forms, surviving relatively unchanged through thousands of years of economic evolution. Just as in my discussion of DNA, I argue that evolutionary processes lead to economic institutions that tend to simplify the computational problems facing economic agents, since a "complicated institution" that imposes huge computational burdens on its participants tends to reduce their payoffs and thereby making it less likely to persist. This is why I believe that the impossibility results reviewed in section 2 that are based on the mere *existence* of artificial worst case games or institutions that force agents to solve computationally intractable problems are irrelevant for thinking about the question whether agents are capable of finding rational/optimizing solutions to the problems they typically face. The relevant question is whether *actual institutions* require agents to solve intractable computational problems.

Why have certain market institutions survived relatively unchanged after many centuries of economic evolution? I argue that their success is not only due to the fact that they are *allocationally efficient,* but also due to the fact that that they are *computationally efficient* in a sense I attempt to explain below. Consider a particular market institution, the double auction (DA) market, an idealization of an ancient market institution where buyers and sellers of a homogeneous commodity meet at a common "trading post" to conduct exchange. The DA market opens at specific times, during which traders dynamically announce bids and asks for various quantities of the good and transactions are consummated at an endogenously determined set of transaction prices. The sequence of transaction prices are determined when holders of the "best" current bid and asks (i.e.

[19] Recent advances in communication and the storage and dissemination of information have greatly accelerated the rate of "social evolution" relative to natural evolution. In nature, when a genetic mutation in DNA yields a better blueprint for a more successful organism, it may take many generations and hundreds or thousands of years for the new genetic innovation to propagate throughout the species. However the rapid rate of communication in human societies enables new innovations to transform and simplify the lives of billions of humans in the span of a few decades or even years.

highest bid and lowest ask) decide to trade: the actual transaction price is determined by a variety of rules, but is typically some price in the interval between the current bid and ask. Modern commodity markets such as the Chicago Board of Trade continue to use specialized forms of the DA institution, and the basic institution perseveres even though new communications and computer technologies are rapidly leading to electronic versions of the DA market that are replacing traditional oral double auction conducted in a frenzied "trading pits". Theoretical analyses of the DA market focus on its role in endogenous price formation since these markets have no central "Walrasian auctioneer" whose goal is to clear the market. Extensive experimental evidence shows that transaction prices rapidly converge to values close the the competitive equilibrium (CE) predictions, and observed allocations are nearly 100% efficient, even in relatively "thin" markets with few buyers and sellers.

Economic theorists model the DA market as a non-cooperative continuous-time game of incomplete information: their goal is to "explain" observed outcomes as realizations of Bayesian Nash equilibria of these games. Unfortunately none of these game-theoretic models have analytic solutions, and the problem of computing approximate equilibria to these games appears to be a computationally intractable since equilibria require solving continuous-time, continuous-state dynamic programming problems for the agents' optimal trading strategies and as I discussed in section 2 even single agent DP problems can be shown to be intractable in the worst case using deterministic algorithms. As a result economic theory provides us with very few specific predictions of trading behavior and overall outcomes in DA markets. So in what sense is the DA institution computationally efficient?

The apparent intractability of solving game-theoretic models of the DA does not square with considerable experimental evidence that demonstrates that even untrained undergraduates are able to quickly learn to trade in DA markets and observed outcomes are very close to the predicted competitive equilibrium outcomes. Research by Gode and Sunder (GS), (1993) and Rust, Miller and Palmer (RMP) (1993) suggests a potential resolution to this paradox: *outcomes in the DA market are typically "collectively rational" even if some traders are not "individually rational".*[20] This "robustness property" of the DA institution vastly simplifies the computational problems of traders. Instead of solving very complicated dynamic programming problems that require keeping track of the entire history of bids and asks of other traders and doing Bayesian updating of beliefs

[20] For example GS studied DA markets populated by "zero intelligence" (ZI) traders whose bids and asks are simply *IID* draws from uniform distributions truncated at the reservation value to enforce only the minimal requirement that the ZI trader never trades for negative profit. GS showed that collections of ZI traders display collective rationality in that price trajectories converge to CE outcomes with probability close to 1 and realized allocations are nearly 100% efficient. RMP 1993 showed that similar outcomes also emerged with collections of heterogeneous strategies, many of which an easily exploit the vulnerability of the ZI strategies. The winning strategy in their computerized double auction tournament was a simple rule of thumb which appears to be a highly successful heuristic for a wide range of environments, and which performed well in experiments with human opponents.

about the valuations of other individual traders, the rapid convergence to CE outcomes implies that transaction prices become good sufficient statistics and the expected profit maximizing strategy becomes very simple: a trader need only compare previous transaction prices to their reservation value and, if positive, place a bid or ask close to previous transaction in order to guarantee a high probability of trading and realizing their profit.

Although it is true that rational traders can potentially make "excess profits" if they succeed in identifying and exploiting irrational noise traders, the results of RMP suggest that the combination of the high levels of noise and limited amounts of data in individual DA trading sessions make it very difficult to identify strategies employed by individual opponents, so that it is unlikely that rational traders could identify and exploit irrational traders unless they were repeatedly trading with these agents in the same market. Furthermore the evolutionary computer tournaments conducted in Miller, Palmer and Rust (1993) show that the irrational traders are rapidly driven into extinction anyway, so in the absence of a steady stream of noise traders the long-run outcomes of DA markets are dominated by the more intelligent or "rational" traders. Unfortunately, given the apparently intractable computational problem of solving for Bayesian Nash equilibria directly, we don't know whether these long-run outcomes in the DA market are close approximations to the game-theoretic predictions. However the basic point here is that the combination of the knowledge capital embodied in DA trading rules and evolutionary forces shaping long-run trading strategies in the DA market seem to greatly reduce the computational burdens on individual traders while leading to collectively rational outcomes, i.e. prices and efficiency levels that are very close to those predicted by (full information) models of competitive equilibrium.[21]

There are a wide variety of other market institutions in addition to the DA market. In general economists believe that these market institutions generally represent very good solutions to the economic allocation problem that they were "designed" to solve. The reasons for this have been extensively analyzed in economic theory. In the next section I wish to explore a new hypothesis about their success and stability: namely, that markets succeed in solving extremely complex, potentially intractable resource allocation problems via "decentralization" and decentralization is "computationally efficient".

---

[21] For further detail on the institutions, theory and empirical evidence on the performance of the DA market, see Friedman and Rust, 1993.

4. Is Decentralization "Computationally Efficient"?

The previous section offered some hardware and software solutions to the problem of economic complexity. I think that these ideas will help us greatly extend the class of models we can solve. But I don't think these ideas will be enough in themselves. I this section I present some speculations about the role of *decentralization* as an additional method for solving complex problems. I view decentralization as an interface between hardware and software, a type of "operating system" that constitutes the most efficient method for performing large-scale computations on massively parallel hardware. My argument for the computational efficiency of decentralization consists of two parts. First, I argue that there are physical/technological reasons why nature's most powerful computers take the form of massively parallel processors with millions or even billions of individual processors. Second, I conjecture that there are mathematical/logical reasons why nature relies on decentralization as the operating system for managing and controlling massively parallel systems. Paradoxically, my argument for the computational efficiency of decentralization is based on an "impossibility result," namely, the limitations of using centralized methods to manage large scale systems. These limitations are due to the informational problems stressed by von Hayek and to problems of intractability stressed in the literature on computational complexity.

The first issue is to define what we mean by "decentralization". Unfortunately Malinvaud's (1990) entry in the New Palgrave dictionary does not attempt to define this concept, and while the definition in the Oxford English dictionary conveys the basic idea, it is not sufficiently precise for our purposes. [22] It turns out to be quite difficult to provide a precise mathematical definition of decentralization, although the intuitive notion is quite clear: a decentralized system is one which has no identifiable "center" that controls the behavior or dynamics of the individual *agents* (i.e. processors, particles, consumers, firms, etc.) comprising the system. Instead, control and information processing in decentralized systems is distributed among the agents comprising the system and these agents are autonomous in the sense that their behavior or laws of motion are governed primarily by their own "objective functions," although their objectives may be affected by messages, competition, or other types of interactions with other agents in the system. [23] A

[22] The definition of decentralization is defined as the opposite of centralization, that act of *centralizing* which is "to concentrate (administrative powers) in a single head or centre, instead of distributing them among local departments."

[23] There is a closely related concept of "informational decentralization" arising from the literature on resource allocation mechanisms (e.g. Hurwicz 1972 and Mount and Reiter 1974) in which information needed to make resource allocation decisions is not directly available in one place for use by a central planner, but rather is distributed throughout the economy in the form of private information about economic agents' endowments, preferences, technologies, etc. The objective of this literature is to determine whether it is possible to design "game-forms" and "message processes" that "implement" various allocations or social choice rules. The equilibria of these games can be thought of as decentralized methods for solving social planning problems since they eliminate the need for a central planner.

final requirement that distinguishes a decentralized system from an arbitrary collection of particles, processors, or agents is that the system as a whole solves an identifiable problem. Thus, even though the individual elements in the system apear to behave autonomously (and may not even be “aware” that they are part of a larger system), the collective outcome of the interactions appears to an outside observer as if the particles had somehow cooperated to solve the overall problem, a phenomenon that is also referred to as *emergent computation.* In order to make this intuitive definition of clearer, it is helpful to illustrate it via several examples of emergent computation via decentralization of massively parallel systems.

Soap films. This is perhaps one of the simplest examples of a decentralized system. If you dip a wire loop into a bucket of soapy water, a soap film will typically form across the loop when it is extracted from the bucket. This soap film is the solution to a global optimization problem, namely how to arrange the trillions of soap and water molecules in the film into an energy-minimizing configuration. A physicist (e.g. Richard Palmer, private communication) will tell you that there is no “central planner” that solves this global optimization problem: instead the shape of the soap film is an emergent computation that arises endogenously from the collective interactions of the soap and water molecules, whose individual objective function are to find an energy-minimizing positions in the electrostatic field created by neighboring molecules. It is important to note that this decentralized system solves the global optimization problem extremely rapidly even though an analysis of the worst case complexity of global optimization suggests that the problem is intractable, and numerical experience with alternative methods such as solving systems of partial differential equations describing the dynamics of the soap film lead to intractable computational problems. For this reason mathematicians studying these types of PDE’s typically resort to simply dipping wire loops into buckets of soapy water, as this seems to be the fastest “computer” for studying the effects of different boundary conditions. This example shows that there exist decentralized, massively parallel analog computers guided by thermodynamic or electrodynamic principles that can rapidly solve what would otherwise seem to be an intractable global optimization problem.

The immune system. This is a massively parallel system whose individual elements include “various effector ‘combat troops’ (such as macrophages, antibodies of various specificities and isotypes and cytotoxic T-cells) as well as signal corps (exemplified by helper T cells and their cytokines)” (Segel, 1996, p. 2). These elements have well defined objectives functions: for example “T cells bear receptor molecules that that recognize an MHC-protein combination that matches the special shape (specifity) of the receptor. Suitable recognition triggers an intracellular molecular cascade that leads to killing by the cytotoxic T cells (and to signalling by the T helpers).” (Segel, 1996, p. 2). The immune system is decentralized, according to Segel, because it “is entirely distributed (to a first approximation)", yet it "has a fairly well defined major mission, the destruction of pathogens".

The brain. This is a massively parallel system whose individual elements are neurons that communicate via electrical nerve impulses and secretion of various neurotransmitters and neuromodulators the synapses connecting neurons. Although there is an obvious hierarchical large scale structure to the brain with various specialized processing areas (visual cortex, hippocampus, cerebellum, thalamus, etc.) that is probably largely due to the "central planning" provided by DNA during initial stages of development, the operation of most of the adult brain appears to be best described as a decentralized system. The reason is that there is no identifiable "homonculus" or central planner that coordinates the activities of all the nerve cells:

> "The anatomy of frontal cortex and other areas beyond the primary sensory areas suggests an information organization more like an Athenian democracy than a Ford assembly line. Hierarchies typically have an apex, and following the analogy, one might expect to find a brain region where all sensory information converges and from which motor commands emerge. It is a striking fact that this is false of the brain. Although there are convergent pathways, the convergence is partial and occurs in many places and occurs in many places many times over, and motor control appears to be distributed rather than vested in a central command center. (Churchland and Sejnowksi, 1992, pp 24–25)

Individual neurons in the brain appear have their own objective functions: in very rough terms, they seek stimulation and avoid inhibition (see, e.g. Klopf, 1982). A number of neuroscientists and computer scientists have developed theories of cognition that hypothesize that the brain operates as a sort of "society" or competitive economy:

> How can intelligence emerge from nonintelligence? To answer that, we'll show that you can build a mind from many little parts, each mindless by itself. I'll "Society of Mind" this scheme in which each mind is made of many smaller processes. These we'll call *agents*. Each mental agent by itself can only do some simple thing that needs no mind or thought at all. Yet when we join these agents in societies — in certain very special ways — this leads to true intelligence. (Minsky, 1986, p. 17).

However it is obvious that at any given time the brain as a whole as well defined objectives such as eating, work, sex, etc. Furthermore it is also clear that communications within the brain are asynchronous, depending on what often appears to be random intervals between firing of adjacent neurons and secretion of neurotransmitters and inhibitors in the synapses.

The Economy. A competitive economy is a final type of hardware with these characteristics. The individual processors are the human agents who interact in the economy. A competitive economy is decentralized because there is no central planner that directs the activity of the agent: instead agents behave autonomously, setting prices and making production and consumption decisions in order to maximize individual profit or utility. The "objective" of the economy as a whole is less clear than in the previous two examples, but economic theory teaches us that the outcome is an efficient allocation of resources: i.e. under ideal conditions economy operates as if it were maximizing a weighted sum of individual utilities.

A previous version of this paper hazarded a conjecture that decentralization is computationally efficient, but based on feedback received from the initial draft, I realized that I was unable to provide a precise mathematical definition of "decentralization" much less prove that it is computationally efficient. The intuitive definition of a decentralized system is one that is not under central control. However what if a central planner "mimics" a decentralized process? For example, in an economic context a central planner might attempt to mimic a decentralized outcome by using "shadow prices" and acting like a Walrasian auctioneer, broadcasting a sequence of trial prices and allowing agents to report their preferred consumption bundles until a price vector p^* is found with the property that the sum of the reported "notional" excess demands is zero. It is not obvious whether this price adjustment process should be regarded as "centralized" or "decentralized". I believe a key characteristic distinguishing centralized from decentralized systems is the level of synchronization and autonomy of the individual agents: decentralized systems are ones in which agents have substantial autonomy and operate asynchronously. Unfortunately it is hard to think of a simple definition that enables me to draw a clear cut dividing line distinguishing centralized and decentralized systems that everyone would agree upon. However I it is possible to provide case by case classifications of centralized and decentralized algorithms: it is a situation of "you know it when you see it". I conclude this section by providing a number of examples of centralized and decentralized algorithms for computing competitive equilibria, and discuss some of the computational advantages of decentralized algorithms.

4.1 Examples of Centralized Algorithms

A classic example of a centralized method for computing CEs is the "programming approach" suggested by Negishi and described in Dixon and Parmenter, 1996, section 2.1. In the programming approach, one repeatedly maximizes a weighted average of individual utilities subject to aggregate resource constraints. The (normalized) Lagrange multipliers for the resource constraints serve as "shadow prices" that can be regarded as trial values of the competitive equilibrium price vector. The welfare weights are adjusted and the SA problem (3.1) and (3.2) is repeatedly re-solved until there exists a set of welfare weights $(\hat{a}_1, \ldots, \hat{a}_K)$ and corresponding shadow prices $\hat{\lambda}$ such that $\hat{\lambda} x_i = \hat{\lambda} e_i$ holds for each consumer i. It is easy to show that any such solution is also a CE for the economy given initial endowments e_i, $i = 1, \ldots, K$.

I believe that a central planner that mimics a Walrasian auctioneer is also a centralized process, and for the same reason the classic tatonnement process is also a centralized algorithm because a) the auctioneer controls the sequence of trial prices, b) agents are not allowed to consume trial consumption bundles, and c) agents' reports must be truthful and are completely synchronized.

There are versions of tatonnement that I would describe as "less centralized" and these algorithms suggest there may be a large gray area where disagreement would start to arise about whether the algorithm is centralized or decentralized. An example is the recent "Walras algorithm" proposed by Cheng and Wellman (1996) which can be regarded as an asynchronous, distributed version of of the standard tatonnement process but with d separate auctioneers for each of the d goods in the economy. The algorithm operates in discrete time and at each time t and for each market i, each agent k is required to transmit a demand *function* $x_k(p_i, p_{-i}^{t-1})$ to the auctioneer in market i, where p_{-i}^{t-1} denotes the (fixed) prices of all commodities except i generated from the Walras algorithm in the previous period. Auctioneer i then adds the transmitted demand functions to compute an aggregate excess demand function for market i taking the prices of other commodities as fixed. The auctioneer then quotes a new price p_i^t that sets the excess demand in market i equal to zero, *independently of the price adjustments made by the auctioneers in the other* $d-1$ *markets*. Even though auctioneers ignore the feedback effects of their price adjustments on the other markets, Cheng and Wellman prove that if commodities are *gross substitutes* (a sufficient condition for global stability of ordinary tatonnement and other continuous price adjustment processes) the Walras algorithm generates a stochastic price sequence $\{p^t\}$ that converges to the CE price vector p with probability 1. Although the Walras algorithm is less centralized (or more decentralized) than the standard tatonnement process, the method still assumes that the d auctioneers in each market are virtually equivalent to central planners and for this reason I view it as a centralized approach.

Almost all other methods that have been proposed by economists and operations researchers to compute approximate CE including variations of Newton's or "Johansen's method" (see Dixon and Parmenter, 1996), or Scarf's algorithm (see, Scarf, 1982), or other more elaborate methods based on solving variational inequalities or sequences of linear or nonlinear complimentarity problems (see Nagurney, 1996 or Rutherford, 1985) are algorithms that can be interpreted as requiring strict coordination, synchronization, and reporting requirements on the part of individual agents, and therefore must be regarded as centralized approaches to solving the SA problem. There are other examples of algorithms for computing equilibria have been introduced that are explicitly referred to as "decentralized" (e.g. Reiter, 1994) but closer inspection reveals that the algorithms are "informationally decentralized" but not necessarily "computationally decentralized" in the sense I am trying to convey here. In particular, Reiter's algorithm appears to be inherently serial rather than parallel (due to the fact that agents are required to act sequentially, with agent i being unable to act until receiving a message from agent $i-1$), and the complexity of the calculations required of each agent is $\Theta(K)$ rather than $O(1)$, violating the property of computational constant returns to scale.[24]

4.2 Examples of Decentralized Algorithms.

I already discussed several examples of "decentralized algorithms" for computing approximate CE in my discussion of the double auction market in the section on knowledge capital: the work of Gode and Sunder 1993 and Rust, Miller and Palmer 1993. Another example of a decentralized algorithm is the the bilateral bargaining model of general equilibrium price formation of Axtell and Epstein 1996.[25] The Axtell and Epstein paper is a computational simulation of exchange and price formation in which a population of agents with preferences over continuous d-dimensional commodity bundles are randomly matched and are allowed to engage in bilateral bargaining. Although their environment is "classical" and approximate equilibria could have been calculated by standard centralized solution methods reviewed in the previous section, their computer simulations demonstrate that a completely decentralized process of repeated bilateral trade between randomly-selected pairs of agents, whether done sequentially or in parallel, results in an approximate CE solution appearing as an emergent computation. They also find that solutions emerge quite rapidly:

> "Bilateral trade processes are meaningfully viewed as a kind of distributed computation of near-Pareto optimal allocations. It is argued that such distributed trade processes provide more faithful interpretations of the 'invisible hand' than does the first welfare theorem of Walrasian general equilibrium. . . . It is shown that the number of bilateral interactions required to reach Pareto optimality is linear in the number of agents and the number of commodities. It is argued that the non-polynomial complexity of algorithms for the computation of Walrasian equilibria make them unrealistic as metaphors for real markets." (Axtell and Epstein, 1996, p. 1)

The Axtell and Epstein model can be thought of a computational implementation of Gale's (1986) theoretical model which proves that Walrasian outcomes are the limit of a model of random matching and bargaining. I classify this as decentralized because a) agents bargain over goods, and thus have the right to propose and reject trial prices, b) agents can choose how long to bargain and when to consume, and c) the matching and bargaining is asynchronous and typically done in parallel.

---

[24] Part of this result is due to the fact that in Reiter's model the number of "choices" made by each agent is equal to the total number of agents in the economy. In our notation, this is equivalent to assuming that $d = K$.

[25] I do not attempt to survey the growing literature on decentralized, agent-based approaches to computation. Other examples of work in this area include Albin and Foley (1992), Bell (1996), and Testfatsion (1996). The agent-based approach has also been used to compute approximate equilibria of dynamic games. See, for example, Marimon, McGrattan and Sargent (1990) and Pakes and McGuire (1996).

An important goal of agent-based computational modeling is to understand how a large computational task can be solved via the cooperative efforts of a large number of agents (processors) in a loosely coupled (or asynchronous) and decentralized fashion. The recent literature on "emergent computation" in computer science is pursuing similar goals. Computer scientists such as Baum (1995), Holland (1976), Huberman *et. al.* (1988), and Waldspurger *et. al.* (1992) have come to recognize the role of competition and "prices" as a powerful means of facilitating and coordinating emergent computation. I believe the key to understanding decentralized algorithms is to understand how to provide the right incentives to each processor and design the right "institutions" to make it easy for each processor to behave cooperatively, each solving a very small piece of the overall problem. Although a number of researchers have already shown that it is possible to build decentralized agent-based models that work well in specific contexts, we still do not have a general theory of how to design agent-based models to solve given problems nor a good understanding of the sense in which these sorts of parallel, decentralized algorithms may be computationally efficient.

However when such institutions are in place the intuition why decentralized methods are more efficient than centralized methods for controlling massively parallel systems is intuitively clear: 1) there is Hayek's observation that there are substantial problems in making sure the central planner always has the sufficient information to make good decisions and with communications costs and delays, it is unlikely that any central planner could ever be kept up to date sufficiently rapidly to take good decisions on a second by second basis, 2) there are complexity problems associated with determining how to break up each given problem into little subtasks, assign them to individual processors and reassemble the solutions to the subproblems into an overall solution to the problem. The fact that there has been very slow progress on developing a general purpose operating system for massively parallel machines like the Connection Machine is an indication of the substantial difficulties involved. Typically a great deal of human insight, intervention and hand coding is required in order to effectively parallelize any particular problem. The hope is that a truly decentralized system would have the intelligence to self-organize and solve each particular problem it confronts using competitive principles perhaps the way the human brain is able to solve a wide variety of different tasks. The challenge is to provide the right institutions so decentralization and competition succeeds in harnessing the power of massively parallel systems rather than resulting in anarchy and chaos, otherwise it is not clear that having larger massively parallel systems is really a significant advantage. Minsky aptly summarized the importance of having effective institutions in his discussion of the evolution of human thought:

> Our ancestors diverged from their relatives, the gorillas and the chimpanzees, only a few million years ago, and our human brains have grown subtantially in only the last few hundred thousand years. . . .

> The evolutionary interval was so brief that that most of our genes and brain structures remain nearly the same of those of the chimpanzees. Was it merely an increase in the brain's size and capacity that produced our new abilities? Consider that, by itself, an increase in the size of the brain might only cause the disadvantage of mental confusion and the inconvenience of a heavier head. However, if we first evolved the ability to *manage* our memories, we could then take advantage of more memory. Similarly inserting new layers of agents into old agencies might only lead to bad results — unless this were preceded by mechanisms for using such layers as "middle-level managers" without disrupting older functions. In other words, our evolution must have worked the other way: first came enhancements in abilities that made it feasible for us to manage larger agencies. Then, once we had the capability for using more machinery, natural selection could favor those who grew more massive brains." (Minsky, (1986) p. 321)

5. Concluding Remarks

This essay turned out to have much more of a philosophical tone than I would have liked, but this is probably inevitable given that the debate over whether there are fundamental limits to knowledge and whether rational/equilibrium behavior is possible cannot be satisfactorily resolved on the basis of purely logical considerations. However my ultimate goal is really quite practical, namely to find new methods to solve larger, more realistic models. In this vein, I think the most compelling "proof" will not come from talking about the issue in any generality, but by actually building realistic large scale models where agents are behaving approximately optimally or approximately in equilibrium.

This essay might be viewed as an attempt to revive Guy Orcutt's "big science" vision for economics via a more decentralized, agent-based modeling approach, and it is likely to be met with considerable skepticism by many economists. A criticism of the "big science" approach is that large scale computer simulations do not necessarily amount to a great deal of "understanding": indeed the sheer complexity and scale of these simulations quickly move us out of the realm of something that we can easily understand. If we acknowledge that even under the best of circumstances all we can expect to get from an agent-based simulation is a computerized replica of many of the complexity and idiosyncracies of real world economies, then what is the point? Why not simply observe the real economy and dispense with the agent-based simulations? My answer is two-fold. First, we want computerized agents and economies to study and predict the effects of hypothetical policies and scenarios, and it is typically much cheaper to discover bad policies in computerized models than to try them out on human "gineau pigs." Second, to have a good "understanding" of any phenomenon requires multiple approaches. Certainly we cannot have understanding without the assistance of some reasonably simplified and abstract theories and "toy models" that give us insight into the essential features and principles underlying the real world. However if this is all that we have, I would not call this "understanding" either. For example much of the theoretical literature in economics is focused on the rather narrow question of proving the existence of an equilibrium for abstract games and model economies. We might fully understand the assumptions and proofs in these models, but do we really think this theory gives us a complete understanding of economics? To have a complete understanding, we need to be able to calculate detailed implications and predictions of these abstract theories and determine whether the predictions of these models are consistent with what we observe in the real world. So, we can't pretend to have a complete understanding of *real economies* until we can show that the detailed implications of our theories provide sufficiently accurate representations of the real world that we could take our models seriously for forecasting and policy analysis. I believe this was Orcutt's vision and it is my vision. This essay has argued against the defeatist view that there are fundamental computational limits that would prevent us from eventually obtaining this sort of understanding, and I have suggested some ideas about how to do this. But in order to make really large breakthroughs I think we will need to structure our computations to more faithfully mimic the computational "solution" that nature and the economy have "discovered" over thousands and millions of years of social and biological evolution. In particular, this means using principles of decentralization to harness the power of massively parallel processors.

I conclude with a motivational example that illustrates how a supposedly intractable problems can be solved via a combination of innovative algorithms, massive parallelism and decentralization. This particular computational breakthrough occurred for the mathematical problem of determining the prime factorization of large integers.[26] There is no known polynomial-time algorithm for prime factorization, although it is not known whether prime factorization is an intractable mathematical problem.[27] However the prime factorization of large integers is generally regarded as such a difficult problem that it has formed the basis of the RSA public key cryptography system that employs a pair of prime numbers as the "secret key" and the "public key" consisting of product of these integers (known as the modulus) and another integer (known as the exponent). To encode a message a user converts the text into an integer which is then raised to power determined by the public key exponent and the remainder computed using the public key modulus and modulo arithmetic. To determine the inverse encoding requires knowledge of the prime factorization of the modulus. In 1977 the inventors of the RSA encryption system encrypted a sentence using a 129 digit modulus and offered a reward to anyone who succeeded in decoding the sentence

[26] Although this is not not an economic problem *per se* it does have serious economic implications since prime factorization underlies the public-key cryptography systems that are currently being considered for implementing secure electronic commerce.

[27] Specifically it is not known whether the prime factorization problem has exponential complexity in the number of digits of the input integer to be factored. It is not even known whether the prime factorization problem is *NP-complete*, although many computer scientists and mathematicians currently believe that it is not.

since their analysis of the computational complexity of the prime factorization problem suggested that it would take over 40 quadrillion years to find the prime factors of their 129 digit modulus using the best known algorithms and computer hardware. In 1994, Bell Communications Research Scientist Arjen Lenstra cracked the RSA code using a massively parallel algorithm known as the *multiple polynomial quadratic sieve* in a period of 8 months using idle CPU cycles of some 1,600 workstations connected over the Internet. Although the total number of calculations required to break the code was quite large (requiring some 100 quadrillion computer instructions), "It was inconceivable 17 years ago that this code could ever be broken" (Lenstra, quoted in Leutwyler, 1994). This example foreshadows one of the main points of this essay: a combination of clever software design, decentralization, and massive parallelism made it possible to solve a problem that was previously thought to be "insoluble".[28]

[28] By the way, the encoded text consisted of the single sentence "The magic words are squeamish ossifrage."

6. References

Albin, P. and D.K. Foley (1992) "Decentralized, Dispersed Exchange without an Auctioneer" *Journal of Economic Behavior and Organization* **18** 27–51.

Anderson, A.R. (ed.) (1964) *Minds and Machines* Prentice-Hall, Inc. Englewood, N.J.

Arrow, K.J. and L. Hurwicz (1958) *Econometrica* **26** 522-552.

Axtell, Robert and Joshua Epstein (1996) "Distributed Computation of Optimal Allocations Through Bilateral Exchange" manuscript, Brookings Institution, Washington, D.C.

Baum, E.B. (1995) "Toward a Theory of Mind as a Lassiez-Faire Economy of Idiots" manuscript, NEC Research Institute, Princeton, New Jersey.

Bell, A. (1996) "Bilateral Trading on a Network: Convergence and Optimality Results" manuscript, Vanderbilt University.

Binmore, K. (1987) "Modelling Rational Players, Part I" *Economics and Philosophy* **3** 179–214.

Board, R. and P. Tinsley (1996) "Smart Systems and Simple Agents: Industry Pricing by Parallel Rules" Finance and Discussion Series 1996-50, Federal Reserve Board, Washington D.C.

Cheng, J.Q. and M.P. Wellman (1996) "The WALRAS Algorithm: A Convergent Distributed Implementation of General Equilibrium Outcomes" manuscript, University of Michigan, forthcoming, *Computational Economics.*

Chow, C.S. and Tsitsiklis, J.N. (1989) "The Complexity of Dynamic Programming" *Journal of Complexity* **5** 466–488.

Davis, P.J. Rabinowitz, P. (1975) *Methods of Numerical Integration* Academic Press, New York.

Dixon, P.B. and B.R. Parmenter (1996) "Computable General Equilibrium Modelling for Policy Analysis and Forecasting" in H. Amman, D. Kendrick and J. Rust (eds.) *Handbook of Computational Economics* volume 1, 3–85.

Joshua M. Epstein and Robert Axtell (1996) *Growing Artificial Societies* Brookings Press, Brookings Institution, Washington, D.C.

Friedman, E. J. and S. S. Oren (1995) "The Complexity of Resource Allocation and Price Mechanisms Under Bounded Rationality" **6** 225–250.

Gale, D. (1986) "Bargaining and Competition part I: Characterization" *Econometrica* **54-4** 785–806.

Gode, D. K. and S. Sunder (1993) "Lower Bounds on the Efficiency of Surplus Extraction in Double Auctions" in D. Friedman and J. Rust (eds.) *The Double Auction Market: Institutions, Theories and Evidence* Volume XIV in the Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, Massachusetts.

Hayek, F.A. (1945) "The Use of Knowledge in Society" *American Economic Review* **35-4** 519–530.

Hayek, F.A. (1989) "The Pretence of Knowledge" (reprint of 1974 Nobel Prize Memorial Lecture) *American Economic Review* **79-6** 3–7.

Hirsch, M., C. Papadimitriou and S. Vavasis (1989) "Exponential Lower Bounds for Finding Brouwer Fixpoints" *Journal of Complexity* **5** 379–416.

Huberman, B.A. and T. Hogg (1988) "The Behavior of Computational Ecologies" in B.A. Huberman (ed.) *The Ecology of Computation* Amsterdam, Elsevier, 77–115.

Hurwicz, L. (1972) "On Informationally Decentralized Systems" in R. Radner and C.B. McGuire (eds.) *Decision and Organization (Volume in Honor of Jacob Marschak)* 297–336, Amsterdam, North Holland.

Klee, V. Minty, G.J. (1972) "How Good is the Simplex Algorithm?" in O. Shisha (ed.) *Inequalities III* 159–175.

Klopf, A.H. (1982) *The Hedonistic Neuron* Hemisphere Publishing Corporation, New York.

Lange, O. (1967) "The Computer and the Market" in C.H. Feinstein (ed.) *Socialism, Capitalism and Economic Growth* Cambridge University Press, 158–161.

Leutwyler, K. (1994) "Superhack" *Scientific American* **271-1** 17–20.

Lewis, A.A. (1985) "On Effectively Computable Realizations of Choice Functions" *Mathematical Social Sciences* **10** 43–80.

Lewis, A.A. (1985) "The Minimum Degree of Recursively Representable Choice Functions" *Mathematical Social Sciences* **10** 179–188.

Lewis, A.A. (1986) "Structure and Complexity: The Use of Recursion Theory in the Foundations of Neoclassical Ecomomics and the Theory of Games" manuscript, Cornell University Department of Mathematics.

Lewis, A.A. (1992a) "Some Aspects of Effectively Constructive Mathematics that are Relevant to the Foundations of Neoclassical Mathematical Economics and the Theory of Games" *Mathematical Social Sciences* **24**

Lewis, A.A. (1992b) "On Turing Degrees of Walrasian Models and a General Impossibility Result in the Theory of Decision Making" *Mathematical Social Sciences* **25**

Marimon, R. McGrattan, E. and T. Sargent (1990) "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents" *Journal of Economic Dynamics and Control* **14-2** 329–374.

Minsky, M. (1986) *The Society of Mind* Simon and Schuster, New York.

Mount, K.R. and S. Reiter (1974) "The Informational Size of Message Spaces" *Journal of Economic Theory* **8 -3** 161–192.

Nachbar, J.H. (1993) "On Computability in Infinitely Repeated Discounted Games" Washington University, manuscript.

Nagurney, A. (1996) "Parallel Computation" in H. Amman, D. Kendrick and J. Rust (eds.) *Handbook of Computational Economics* Amsterdam, Elsevier, 335–404.

National Science Foundation (1993) *From Desktop to Teraflop: Exploiting the U.S. Lead in High Performance Computing* Report of NSF Blue Ribbon Panel on High Performance Computing, National Science Foundation, Washington, DC 20550.

Nemirovsky, A.S. and D.B. Yudin (1983) *Problem Complexity and Method Efficiency in Optimization* Wiley, New York.

Papadimitriou, C.H. (1994) "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence" *Journal of Computer and Systems Sciences* **48** 498–532.

Paskov, S.H. (1993) "Average Case Complexity of Multivariate Integration for Smooth Functions" *Journal of Complexity* **9** 291–312.

Paskov, S.H. and J.F. Traub (1995) "Faster Valuation of Financial Derivatives" *Journal of Portfolio Management.*

Penrose, R. (1987) *The Emporer's New Mind* New York, Penguin.

Reed, M.A. C. Zhou, J Muller, T.P. Burgin and J.M. Tour (1997) "Conduction of Single Molecules" *Science*

Reiter, S. (1994) "A Decentralized Process for Finding Equilibria Given by Linear Equations" *Proceedings of the National Academy of Sciences* **91** 7169–7173.

Rust, J. Miller, J. and R. Palmer (1993) "Behavior of Trading Automata in a Computerized Double Auction Market" in D. Friedman and J. Rust (eds.) *The Double Auction Market: Institutions, Theories, and Evidence* Volume XIV in the Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, Massachusetts.

Rust, J. (1997) "Using Randomization to Break the Curse of Dimensionality" *Econometrica* **65-3** 487–516.

Rutherford, T.J. (1985) "MPS/GE User's Guide" Department of Operations Research, Stanford University.

Scarf, H. (1982) "The Computation of Equilibrium Prices: An Exposition" in K.J. Arrow and M.D. Intrilligator (eds.) *Handbook of Mathematical Economics* Volume II, Amsterdam, Elsevier, 1007–1061.

Segel, L.A. (1997) "The Immune System as a Prototype of Autonomous Decentralized Systems" manuscript, Weizmann Institute of Science, Rehovot, Israel.

Sikorski, K. (1984) "Optimal Solution of Nonlinear Equations Satisfying a Lipschitz Condition" *Numerische Mathematik* **43** 225–240.

Sikorski, K. (1985) "Optimal Solution of Nonlinear Equations" *Journal of Complexity* **1** 197–209.

Simon, H.A. and A. Ando (1961) "Aggregation of Variables in Dynamic Systems" *Econometrica* **29** 111–138.

Simon, H.A. (1996) *The Sciences of the Artificial* MIT Press, Cambridge, Massachusetts.

Stix, G. (1995) "Toward 'Point One"' *Scientific American* **272-2** 90–95.

Traub, J.F. (1994) "Breaking Intractability" *Scientific American* **270-1** 102–107.

Traub, J.F. Wasilkowski, G. and H. Woźniakowski (1988) *Information-Based Complexity* New York, Academic Press.

Tsitsiklis, J.N. and M. Athans (1985) “On the Complexity of Decentralized Decision Making and Detection Problems” *IEEE Transactions on Automatic Control* **AC-30-5** 440–446.

Waldspurger, C.A. T. Hogg, B.A. Huberman, J.O. Kephart and W.S. Tornetta (1992) “Spawn: A Distributed Computational Economy” *IEEE Transactions on Software Engineering* **18-2** 103–117.

Wasilkowski, G. and H. Woźniakowski (1996) “On Tractability of Path Integration” *Journal of Mathematical Physics* **37-4** 2071–2088.