Has dynamic programming improved decision making?[*]

John Rust, *Georgetown University*[†]

August 22, 2018

Abstract

Dynamic programming (DP) is an extremely powerful tool for solving a wide class of sequential decision making problems under uncertainty. In principle, it enables us to compute *optimal decision rules* that specify the best possible decision to take in any given situation. This article reviews developments in DP and contrasts its revolutionary impact on economics, operations research, engineering, and artificial intelligence, with the comparative paucity of real world applications where DP is actually used to improve decision making. I discuss the literature on numerical solution of DPs and its connection to the literature on reinforcement learning (RL) and artificial intelligence (AI). Despite amazing, highly publicized successess of these algorithms that result in superhuman levels of performance in board games such as chess or Go, I am not aware of comparably successful applications of DP for helping individuals and firms to solve real-world problems. I point to the fuzziness of many real world decision problems and the difficulty in mathematically formulating and modeling them as key obstacles to wider application of DP to improve decision making. Nevertheless, I provide several success stories where DP has demonstrably improved decision making and discuss a number of other examples where it seems likely that the application of DP could have significant value. I conclude that "applied DP" offers substantial promise for economic policy making if economists can let go of the empirically untenable assumption of unbounded rationality and try to tackle the challenging decision problems faced every day by individuals and firms.

Keywords actor-critic algorithms, Alpha Zero, approximate dynamic programming, artificial intelligence, behavioral economics, Bellman equation, bounded rationality, curse of dimensionality, computational complexity, decision rules, dynamic pricing, dynamic programming, employee compensation, Herbert Simon, fleet sizing, identification problem, individual and firm behavior life-cycle problem, locomotive allocation, machine learning, Markov decision processes, mental models, model-free learning, neural networks, neurodynamic programming, offline versus online training, optimal inventory management, optimal replacement, optimal search, principle of decomposition, Q-learning, revenue management, real-time dynamic programming, reinforcement learning, Richard Bellman, structural econometrics, supervised versus unsupervised learning

[*] **Acknowledgements:** I am grateful for financial support from the Gallagher Family Chair in Economics at Georgetown University and helpful comments from Larry Kotlikoff, David Romer, Stephen Rassenti, Kevin James, Dustin Tracy, Preston McAfee, Benjamin Golub, Warren Powell, Amit Gandhi, Hal Varian and Patrick Bajari.

[†] Department of Economics, Georgetown University, e-mail: jr1393@georgetown.edu

> *As new mathematical tools for computing optimal and satisfactory decisions are discovered, and as computers become more and more powerful, the recommendations of normative decision theory will change. But as the new recommendations are diffused, the actual, observed, practice of decision making in business firms will change also. And these changes may have macroeconomic consequences. For example, there is some agreement that average inventory holdings of American firms have been reduced significantly by the introduction of formal procedures for calculating reorder points and quantities.* Herbert Simon, 1978 Nobel lecture

1 Introduction

The term "dynamic programming" (DP) was coined by Richard Bellman in 1950 to denote the recursive process of backward induction for finding optimal policies (or decision rules) to wide class of dynamic, sequential decision making problems under uncertainty.[1] Bellman claimed he invented the term to hide "the fact that I was really doing mathematics inside the RAND Corporation" (Bellman (1984), p. 159), but the actual motivation for the development of DP was eminently *practical.* The earliest applications of DP included Massé (1944) (management of hydroelectric reservoirs), Arrow, Harris, and Marshak (1951) (optimal inventory policy), and Wald (1947) (development of optimal sequential statistical decision rules). However according to Bellman's coauthor and colleague Stuart Dreyfus, Bellman had actually

> "cast his lot instead with the kind of applied mathematics later to be known as operations research. In those days applied practitioners were regarded as distinctly second-class citizens of the mathematical fraternity. Always one to enjoy controversy, when invited to speak at various university mathematics department seminars, Bellman delighted in justifying his choice of applied over pure mathematics as being motivated by the real world's greater challenges and mathematical demands." Dreyfus (2002)

DP emerged as a fundamental tool of applied mathematics, and it revolutionized the way we do economics. It is probably the single most fundamental tool underlying game theory, macroeconomics and microeconomics (see, e.g. Stokey and Lucas (1989), Adda and Cooper (2003), Maskin and Tirole (2001)). As a result, the majority of the modern economics literature can be regarded as a type of "applied DP." However economists use DP primarily as an *academic modeling tool,* but compared to the operations research and engineering literatures, economists have contributed less to the *practical application of DP to improve decision making and policy making.* Why?

I believe this is is largely due to the predominant orientation of economics as a *positive* (i.e. descriptive) theory of the behavior of individuals and firms under the assumption of *unbounded rationality.* Most economists are perhaps too comfortable with the assumption that firms maximize expected discounted profits, individuals maximize expected discounted utility, and even that governments behave as benevolent social planners (social welfare maximizers). Most applied policy making by economists concerns macroeconomic stabilization, counteracting market power, or correcting market failures such as external-

[1] I assume readers are familiar with DP, see Rust (2008) for details, history, and background.

ities or incomplete markets. But the prevailing view that individuals and firms are able to solve their own decision problems perfectly well leaves little motivation for economists to use DP as a *normative* tool to improve decision making in practice. As Simon (1992) noted in his Nobel Prize lecture

> "Decision theory can be pursued not only for the purposes of building foundations for political economy, or of understanding and explaining phenomena that are in themselves intrinsically interesting, but also for the purpose of offering direct advice to business and governmental decision makers. For reasons not clear to me, this territory was very sparsely settled prior to World War II. Such inhabitants as it had were mainly industrial engineers, students of public administration, and specialists in business functions, none of whom especially identified themselves with the economic sciences. During World War II, this territory, almost abandoned, was rediscovered by scientists, mathematicians, and statisticians concerned with military management and logistics, and was renamed 'operations research' or 'operations analysis.' So remote were the operations researchers from the social science community that economists wishing to enter the territory had to establish their own colony, which they called 'management science.' The two professional organizations thus engendered still retain their separate identities, though they are now amicably federated in a number of common endeavors." (p. 349–350).

Simon questioned the relevance of economists' assumption of unbounded rationality and he noted that "by the middle 1950's, a theory of bounded rationality had been proposed as an alternative to classical omniscient rationality, a significant number of empirical studies had been carried out that showed actual business decision making to conform reasonably well with the assumptions of bounded rationality but not with the assumptions of perfect rationality" (p. 357).

Note that bounded rationality does not imply individuals or firms behave stupidly. The opening quote from Simon suggests that he recognizes that there are very strong competitive and evolutionary advantages to making better decisions. However as I discuss below, my impression is that formal DP has not been widely adopted to improve decision making by individuals and firms. If DP can really help them make better decisions, why hasn't it been more widely adopted in practice?

Three possible answers come to mind: 1) the assumption of unbounded rationality may in fact be a reasonable approximation, so most individuals and firms behave "as if" they solved their DP problems, but via experience, learning, trial and error experimentation, so they do not need to explicitly formulate and solve a DP, 2) individuals and firms are behaving suboptimally, but are not aware of DP or lack the ability to actually formulate and solve their DP problems, or 3) there is a *curse of dimensionality* and the complexity of the real world makes it impossible for *anyone* to formulate and solve DP problems that can sufficiently closely approximate many or most the problems individuals and firms actually confront. I will consider all three explanations in this article, but the most likely explanations for the limited practical application of DP are items 2 and 3 above. Further, if explanation 3) is true, it automatically rules out explanation 1, consistent with Simon's view of the world.

Bellman coined the term "curse of dimensionality" to refer to the exponential growth in computer power required to solve bigger, more realistic DP problems.[2] For many years it was not clear whether some amazing algorithm would be discovered that could break the curse of dimensionality, but subsequent research by computer scientists formalized the curse of dimensionality as an exponential lower bound on the problem's *computational complexity* denoted by comp(ϵ) which is a lower bound on the amount of computer time required to solve a problem with a maximum error of ϵ in the worst case, using *any algorithm.*[3] When the lower bound on comp(ϵ) increases proportionately to $(1/\epsilon)^d$, there is an inherent curse of dimensionality, and computer scientists refer to such problems as *intractable.*[4] There are subclasses of DP problems that can be shown to be tractable (i.e. not subject to a curse of dimensionality), and thus solvable in polynomial time or have complexity bounds that do not increase exponentially in the problem dimension d. For example linear-quadratic DP problems (see e.g. G. Chow (1976)) can be solved in polynomial time. Rust (1997) proved that it is possible to break the curse of dimensionality for *discrete choice DP problems* (DCDP, i.e. problems with only a finite number of possible choices but with d continuous state variables). These problems can be solved using a randomized algorithm (i.e. using Monte Carlo integration) and Rust showed that the complexity of computing an approximate solution to DCDP problem with an expected maximum error of ϵ is bounded above by $(d/\epsilon)^4$, so this class of problems can also be solved in polynomial time. However C. Chow and Tsitsiklis (1989) proved that DP problems with continuous state and decision variables are subject to the curse of dimensionality, at least in the worst case. This implies that if individuals and firms have finite computational capacity, there is *no algorithm* no matter how clever, that can a find a good approximation to the optimal DP decision rule for sufficiently large, realistically formulated DP problems that individuals and firms confront in the real world.

If individuals and firms are boundedly rational (i.e. they have finite computational and information processing capacity), and if most of the decision problems they face are subject to the curse of dimensionality, it will be physically impossible for them to use DP to compute optimal decision rules, at least for sufficiently large, complex problems and one that they are likely to confront on a day to day basis. Si-

[2] The "size" of a DP problem can often be indexed by its dimension d, i.e. the number of state and decision variables in the DP problem. A problem is subject to a curse of dimensionality if the computer time/power required to even approximately solve it increases exponentially in d.

[3] See Traub and Werschulz (1998) for an accessible introduction to computational complexity for continuous mathematical problems.

[4] There is a related $P = NP$ problem in computer science on the complexity of discrete mathematical problems such as the traveling salesman problem (TSP). These are finite mathematical programming problems can be solved exactly, unlike DP problems or other continuous, infinite-dimensional mathematical problems whose solutions can only be approximated to within an arbitrary error threshold ϵ. For discrete mathematical problems the size of the problem can be indexed by an integer n such as the number of cities a salesman must visit in the TSP, and the complexity function can be written as comp(n), and represents the minimal time to solve a problem of size n using any possible algorithm. There is a curse of dimensionality for discrete mathematical problems if the complexity is bounded below by 2^n. This remains a famous unsolved mathematical problem: if $P = NP$, then there is no curse of dimensionality and a huge class of discrete mathematical problems such as TSP can be shown to be solvable in polynomial time. See Karp (1972).

mon coined the term *satisficing* to refer to a range of suboptimal decision rules that individuals and firms actually adopt, and some of these include "rules of thumb" or what computer scientists call *heuristics* that provide decision rules that are regarded as "good enough" in situations where nobody is capable of calculating or even closely approximating an optimal DP-based decision rule.

What room does this leave for the normative application of DP to real world problems? As the opening quote from Simon noted, the power of digital computers have been growing exponentially. There has also been a tremendous amount of research on newer, better algorithms for solving DP problems. This implies that the set of problems that can be approximately solved using DP is steadily growing. Finally, I observe that although the overall decision problems that individuals and firms actually confront are incredibly complex, difficult to formalize mathematically, and perhaps impossible to solve numerically, there are many examples where it is possible to apply the *principle of decomposition* to identify *subproblems* of an overall decision problems that can be sufficiently well formalized, approximated and solved using DP. One easy example is the use of GPS navigation systems to find shortest or fastest routes to a desired destination. I discuss inventory management at a mid size steel company in section 4, where the application of the principle of decomposition is used to approximate the firm's problem as the solution of about 9000 $d = 2$ dimensional DP problems rather than a single simultaneous $d = 18000$ dimensional problem. Even if the overall decision problem is not exactly decomposable into simpler subproblems, it may be a reasonable approximation and form the basis for a nearly optimal overall decision rule.

Simon noted, "decision makers can satisfice either by finding optimum solutions for a simplified world, or by finding satisfactory solutions for a more realistic world. Neither approach, in general, dominates the other, and both have continued to co-exist in the world of management science." (p. 350). The goal of this article is to provide a number of examples where DP has been successfully implemented and adopted in practice, as well as a number of convincing examples where it appears that DP *could* help individuals and firms do better. The examples I present where DP has been applied, or has the potential to be successfully applied, are examples of Simon's first approach to satisficing, namely, the academic approach of finding optimal solutions for a simplified world.

I believe that we are on verge of more rapid and widespread practical application of DP due to a number of key recent developments. The most important of them is the exponential growth in power of digital computers, which, combined with vastly improved electronic communications has radically changed our daily lives. After a long gestation lag (digital computers and DP both emerged in the early 1950s), and after a fair amount of early hype and a number of unsuccessful efforts to develop "artificial intelligence" (AI), we are now witnessing the power of dynamic programming (DP) to improve decision making, especially when it is combined with related tools from statistics and computer science such as machine learning (ML) and reinforcement learning (RL). The most clear-cut successes for DP/AI-inspired algorithms are quite recent and are limited to relatively narrow domains such as board games (chess, Go, Shogi, etc) where DP-inspired algorithms have been able to achieve *superhuman* levels of performance.

For example a team at DeepMind (Silver et al. (2017)) developed the *AlphaGo Zero* algorithm for playing the game of Go and concluded that "Our results comprehensively demonstrate that a pure RL approach is fully feasible, even in the most challenging of domains: it is possible to train to superhuman level, without human examples or guidance, given no knowledge of the domain beyond basic rules. Furthermore, a pure RL approach requires just a few more hours to train, and achieves much better asymptotic performance, compared to training on human expert data." (p. 358). Silver (2017) report similar success for the *Alpha Zero* algorithm for chess and shogi in addition to Go. These successes are all the more remarkable due to the vast number of possible states (board positions) in chess (4.5×10^{46} and Go (2.08×10^{170}). These types of successes and rapid progress in other areas such as robotics may have motivated leading thinkers such as the late Stephen Hawking to warn that "The development of full artificial intelligence could spell the end of the human race."

Hawking's concern may be overblown, at least in the short run. When we look at broader, less well-defined domains for decision making, such as running a company or common life decisions such as choosing a career or a spouse, it is currently hard to imagine any algorithm — even one that is based on DP — that we would entrust to make better decisions than ones we would make ourselves. For the forseeable future the human neural network seems much better than any artificial neural network in fuzzy situations that require weighing many hard to quantify subjective factors and considerations. This article considers whether individuals and firms can benefit from the formal application of DP by discussing several real-world applications. Following Simon and the literatures on bounded rationality and behavioral economics, I find little evidence to support the standard assumption of unbounded rationality, i.e. that all individuals and firms have unbounded levels of knowledge, rationality and computational ability, and can be modeled as perfect utility or profit maximizers. The preponderance of evidence suggests that individuals and firms behave suboptimally, and sometimes make *bad decisions* that can have serious long term consequences (e.g. panicking and liquidating your stock portfolio at the bottom of a temporary stock market crash). Insufficient knowledge, emotion and cognitive bias (see, e.g. Kahneman (2011)), and limited reasoning/computational capacity can cause us to make suboptimal choices, creating a role for decision support tools such as DP to help us do better by providing *recommended decisions* that serve as intelligent defaults or suggestions that can be ignored, similar to the idea of a "nudge" by Thaler and Sunstein (2008).

The most promising domain for application of DP in the short run is to improve firm decision making, particularly in *pricing and revenue management* where there is evidence that computer generated recommended prices have significantly increased revenues and profits. As a result there has been rapid entry and growth in this industry: a report by Markets and Markets (2016) forecasts it will grow at nearly 19% per year, from $9.7 billion in 2015 to $21.9 billion in 2020. I provide other examples where DP appears to have great promise to help firms solve some of their easier, more tractable subproblems and become more profitable. Thus, though there is indeed a curse of dimensionality and many if not most problems confronting individuals and firms are still beyond our ability to effectively formulate and solve, the rapid improvement in computer hardware, software, and algorithms makes it reasonable to predict steady growth in the practical applications of DP in the future, especially in view of the very rapid growth of *business analytics* software and services, which according to the INFORMS publication *Analytics* is growing at nearly 10% per year.

In section 2 I summarize the main numerical methods for solving DP problems, focusing on algorithms that are less famiilar to economists but are well known in the AI and RL literature. Knowledge of the strengths and weaknesses of different numerical methods and different methods for learning and approximating the objective function of the decision maker (DM) and the law of motion for the state variables affecting payoffs is key to understanding the different ways DP has been implemented in practice. In section 3 I survey the use of DP for *individual* decision making. This is a comparatively shorter section because I am aware of only a few real-world applications of DP, and, due to the *identification problem* discussed in section 2, it is much trickier to estimate the structure of an individual's preferences and beliefs, especially if individuals have subjective beliefs that are at odds with standard economic assumptions of *rational expectations.* In section 4 I discuss several success stories for the application of DP by firms, as well as a number of additional academic studies that suggest that DP based strategies for some of a firm's key decision subproblems may help firms increase profits. These results call into question the standard economic assumption of discounted expected profit maximization by unboundedly rational firms. In section 5 I present some conclusions and suggested directions and questions for future research this area. My overall conclusion is that though DP does not appear to be widely applied in practice so far, we are likely to see rapid growth in its application in future years that is likely to propel a more rational "science-based" approach to decision making by firms. DP also has the potential to help improve individual decision making, and ultimately hopefully also government decision making as well.

2 Machine versus human learning approaches to solving DPs

Virtually all realistic DP problems that are used in practical applications need to be solved numerically. In this section I provide an overview of the main approaches for solving DPs: 1) "standard approaches" that require a high degree of human input and programming, and 2) ML/RL algorithms that can learn optimal solutions through experience and trial and error experimentation.[5] There are two senses of the term "learning" is used in the AI and DP literatures: 1) learning (i.e. approximating) the optimal decision rule, and 2) learning the *problem structure* i.e. the DM's reward/objective function and the laws of motion governing state variables in the DP problem and how rewards and states are affected (often probabilistically) by decisions. In strategic situations such as a dynamic game, the DP solution typically also requires learning the decision rules used by other decision makers, which may take the form of probability distributions over possible moves (as in the case of board games such as chess or Go). The DP/control literature uses the term *adaptive control* to refer to situations where there is incomplete knowledge of the structure of the DP problem, and the decision maker attempts to learn this structure while trying to make optimal decisions at the same time, leading to the classic tradeoff between experimentation and exploitation that is captured in problems such as the *multi-armed bandit problem* Gittins (1989).

Though the literature on DP and AI originated around the same time[6] and both depend on the exponential growth in the speed of the digital computer, the earliest work on the modeling and design of algorithms to solve the DPs was fundamentally an outcome of *human learning* that required *human* programming. A recent branch of AI called *machine learning* (ML) focuses on development of algorithms "to give computers the ability to 'learn' (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed." (*Wikipedia*). A key distinction in the literature is whether learning takes place *off line* (i.e. the underlying structure an approximate solutions are computed before the decision rule is used to actually make or recommend decisions) or in *real time* (i.e. the algorithm continuously updates its estimate of the problem structure and refine its estimate of the optimal decision rule while decisions are actually being taken). Some of the real time learning algorithms also have the advantage of being *model-free* that is, they only require the decision maker to observe *realizations of rewards* (which can be obtained from the real time application of the decision rule, or via sufficiently realistic computer simulations) rather than requiring specific functional forms for the underlying preferences, beliefs and laws of motion that constitute the structure of the underlying decision problem. Another key distinction is whether the learning is *supervised* or *unsupervised.* Supervised learning presumes the exis-

[5] See Rust (2017) for a recent survey on numerical solution of DPs with further references and examples.

[6] The work of Wiener (1948) on *cybernetics* is considered to have provided the foundations for the subsequent developments in AI.

tence of an expert decision maker whose behavior can be observed and used to *train* a decision rule, either by simple extrapolation, or by structural estimation methods that infer the preferences and law of motion for state variables (beliefs) from the observed behavior of expert (presumably human) decision makers. Unsupervised learning such as RL "comes into play when examples of desired behavior are not available but where it is possible to score examples of behavior according to some performance criterion." Barto and Dietterich (2004), p. 50.

2.1 Standard DP algorithms: off-line solution using "human learning"

Though DP can solve problems that are not directly intertemporal decision problems (e.g. shortest path problems) and intertemporal decision problems with objective functions that are non-separable functions of the state and decision variables, the most common applications of DP are for intertemporal decision problems that can be formulated as an expectation of discounted sum of payoffs, e.g.

$$V_0(s_0) = \max_{d_0,\dots,d_T} E\left\{\sum_{t=0}^{T} \beta^t u_t(s_t,d_t)|s_0\right\} \tag{1}$$

where t indexes time, $(d_0,d_1,\dots,d_T)$ are the decision variables and $(s_0,s_1,\dots,s_T)$ are state variables.[7] Thus, the application of DP in practice is heavily dependent on the assumptions of *expected utility* and *discounted utility* maximization (or maximization of expected discounted profits in the case of firms), though there is substantial laboratory evidence that questions the relevance of these assumptions for modeling individual behavior, and other evidence suggests the stock values of publicly traded firms do not equal the expected present value of their dividend streams, and this opens up important questions about exactly what objective function firm managers are trying to maximize, if any.[8]

[7] Though there is a theory of intertemporal decision making for decision makers with some types of non time-separable and non-expected utility preferences, mathematical tractability has limited this theory primarily to the case of *recursive utility* with particular types of *certainty equivalent operator* that is a generalization of the conditional expectation operator in standard time and state separable expected discounted utility theory. However the underlying theory of the validity of DP for such problems is not well developed and even basic questions such as existence and uniqueness to the generalized version of the Bellman equation and its relationship to optimal dynamic decision rules for such problems is not fully understood. See Bloise and Vailakis (2018) for recent results on existence and uniqueness of the solution to the Bellman equation for deterministic DP problems with certain types of recursive utility. As they note, "Despite the interest in recursive utility ... concomitant progress in dynamic programming methods has not occurred in recent years." and "when some form of non-expected utility is introduced ... certainty equivalent might not satisfy the additivity property required to establish Blackwell discounting, rendering the Contraction Mapping Theorem unavailable even when utility is otherwise time additive." (p. 119).

[8] The experimental violations of expected utility such as via the famous Allais and Ellsberg paradoxes are well known. A survey of laboratory tests of the discounted utility (DU) model Frederick, Loewenstein, and O'Donoghue (2002) concludes "The DU model, which continues to be widely used by economists, has little empirical support. Even its developers — Samuelson, who originally proposed the model, and Koopmans, who provided the first axiomatic derivation — had concerns about its descriptive realism, and it was never empirically validates as the appropriate model for intertemporal choice. Indeed, virtually every core and ancillary assumption of the DU model has been called into question by empirical evidence called into question by empirical evidence collected in the past two decades. " (p. 393). Research by Campbell and Shiller (1988) and others has found that stock prices and returns are "much too volatile to accord with a simple present value model." (p. 675).

This serves as a warning that practical application of DP to intertemporal problems that maximize the expected value of time-additive objective function such as equation (1) may not be maximizing the right objective. To apply standard DP we need to assume that if the actual objective is not exactly intertemporally separable, then it can at least be sufficiently well approximated by the expectation of a time-additive sum of payoffs. A further simplification, also dictated in interest of mathematical tractability, is that the stochastic law of motion for state variables is a *controlled Markov process* resulting in the most commonly analyzed class of DP problems known as *Markovian decision problems* (MDP). In the absence of the Markov assumption, an optimal decision rule is potentially a function of the entire history of previous state which substantially increases the dimensionality of the decision problem. Since the well known trick of expanding the state space can capture higher order stochastic dependencies, the Markov assumption is relatively innocuous in most applications.

Under the Markovian assumption, the *Bellman equation* provides the standard recursive way of expressing the solution to the optimization problem (1) via DP, using the principle of backward induction which is the standard method for solving finite horizon, non-stationary problems (i.e. problems where u and p may depend on t). For stationary, infinite horizon MDPs (where $T = \infty$ and u_t and p_t are independent of t) the Bellman equation is

$$V(s) = \Gamma(V)(s) \equiv \max_{d \in D(s)} \left[u(s,d) + \beta \int V(s')p(s'|s,d) \right], \tag{2}$$

and the corresponding optimal stationary decision rule $\delta(s)$ is given by

$$\delta(s) = \underset{d \in D(s)}{argmax} \left[u(s,d) + \beta \int V(s')p(s'|s,d) \right]. \tag{3}$$

We define the *structure* of the DM's decision problem as the objects $[\beta, u, p, D]$. A large theoretical and numerical literature has arise showing how, given knowledge of $[\beta, u, p, D]$, it is possible to solve the Bellman equation (2) for V and the optimal decision rule δ from equation (3). We can think of the value function V as providing the appropriate *shadow price* for evaluating the effect of current choices on future payoffs so that the decision to a complicated infinite horizon sequential decision problem reduce to what appears to be an ordinary static optimization problem that determines δ in equation (3).

The Bellman equation can be written abstractly as a fixed point $V = \Gamma(V)$ where Γ is the *Bellman operator* that maps value functions into value functions as defined in (2). Blackwell (1965) established the existence and uniqueness of a solution V to Bellman's equation for stationary infinite horizon MDPs by showing that Γ is a *contraction mapping.* This implies that the most commonly used algorithm for solving MDPs, *value iteration,* converges to V starting from any initial guess V_0: $V_{j+1} = \Gamma(V_j)$.[9]

[9] Iterations continue until $\|V_{j+1} - V_j\| < \varepsilon$ for some desired convergence tolerance ε. Proposition 2.2.1 of Bertsekas (2017) provides computable error bounds on the distance between the final iterate V_{j+1} and the true solution $V = \Gamma(V)$.

Howard (1960) proposed an alternative algorithm he called *policy iteration* that makes beautiful use of the duality between V and δ. Given any guess of an initial feasible policy δ_0, the *policy valuation step* involves solving the linear system $V_{\delta_0}(s) = u(s, \delta_0(s)) + \beta \int V_{\delta_0}(s')p(s'|s, \delta_0(s))$ for the implied value of this policy, V_{δ_0}. Then the *policy improvement step* calculates an improved policy δ_1 using equation (3) but using the value function V_{δ_0} in place of the true value function V. Pollatschek and Avi-Itzhak (1969) proved that policy iteration is mathematically equivalent to solving for V as a solution to the zero $F(V) \equiv V - \Gamma(V) = 0$ using Newton's method, and Howard showed that for finite MDPs (i.e. where the states and decisions can have only a finite number of possible values), policy iteration converges to the true solution (V, δ) in a finite number of steps.

There is a huge literature on different numerical strategies for approximating V and δ for MDPs with continuous state and decision variables, including different strategies for doing the maximization and numerical integration operations in equations (2) and (3), and strategies for approximating (V, δ) which are infinite dimensional objects (i.e. functions of the state variable s that can assume a continuum of possible values).[10] The two basic approaches are to use successive approximations or policy iteration as the "outer" algorithm to find the fixed point $V = \Gamma(V)$ to the Bellman equation, but in problems with continuous state variables to either to create some finite grid over the state space $\{s_1, \ldots, s_N\}$ and calculate (V, δ) by solving the Bellman equation as a finite state problem at the N of grid points and then interpolate for values s that are not on this grid. Others, including Bellman, advocated the use some sort of *parametric approximation* to V such as approximating $V \simeq V_\omega$ where V_ω is an element of a class of functions that depend on a finite vector of coefficients ω such as a *neural networks* (see, e.g. Bertsekas and Tsitsiklis (1996)) or *Chebyshev polynomials* (Judd (1998)). Then ω can be determined by methods such as *non-linear least squares* by, for example, finding a value $\hat{\omega}$ that minimizes the mean squared Bellman equation residual,

$$\hat{\omega} = \underset{\omega}{argmin} \int |V_\omega(s) - \Gamma(V_\omega)(s)|^2 \mu(s), \tag{4}$$

for some probability distribution μ.[11]

In the absence of *special structure* (e.g. the MDP is linear quadratic, or there are only a finite number of possible decisions, etc) there will be an inherent curse of dimensionality that *no algorithm* can circumvent, as we noted in the introduction. Though the software and algorithms are constantly improving and enable us to solve increasingly high dimensional problems (see, e.g. Brumm and Scheidegger (2017)), there are still very many interesting applied problems that are well beyond our ability to solve, and in my opinion

[10] See Rust (2017) and Rust (1996) for surveys of this liteature.

[11] Alternatively the sup-norm can be used and $\hat{\omega} = argmin_\omega \|V_\omega - \Gamma(V_\omega)\|$ where $\|f\| = \sup_s |f(s)|$ though this is often more difficult to implement than least squares, though issues arise over equivalence of the L_2 and sup norm in continuous state problems, i.e. minimizing the L_2 norm does not necessarily insure the sup norm error is small, so $V_{\hat{\omega}}$ may not be close to the true fixed point V which satisfies $\|V - \Gamma(V)\| = 0$.

this constitutes a major reason why DP has not been widely used in practice. Ideas from the ML literature including neural networks, Gaussian process regression, and related methods offer considerable promise to continue to extend the size of DP problems that can be solved (see, e.g. Bertsekas and Tsitsiklis (1996), Powell (2010) and Bilonis and Scheidegger (2017)), but as a mathematical matter the application of ML methods cannot break the underlying curse of dimensionality, so these methods should not be regarded as a panacea. Further, most of the successful applications of DP still rely critically on *human insight and intervention* to recognize special structure in a problem (including the existence of simpler, decomposable subproblems of an overall decision problem), how to choose the functional forms representing the DM's preferences and the laws of motion for the problem, and how to design a numerical algorithm from the many choices available that can best exploit the structure of the problem (for example, the choices of optimization algorithms, approximation method and architectures, methods for numerical integration, etc).

Of course, even if the solution to a DP can be computed relatively accurately, the solution may not be useful for improving decision making if the assumed structure of the problem, say $[\beta, u, p, D]$, differs greatly from the actual structure. This is the other critical aspect where learning is required — how do the external advisors to a DM (who are the ones that formulate and solve the DP) learn about the underlying problem structure that the DM (either a human, firm or organization) actually faces? Though the term *actor-critic algorithms* has a more specific meaning in the RL literature that I discuss below, I refer to it here in a more generalized sense to capture how DP is used in practice to improve decision making. The *actor* is the DM who seeks advice in the form of recommended decisions from a *critic* (or policy advisor) who is presumed to have a comparative advantage in solving the actor's decision problem. The next sections provide examples of firms that play the role of the critic by solving the DP of their clients, the actors, to provide them with recommended optimal decisions.

An important general question is *how does the critic learn the structure of the decision problem that the actor confronts?* In economics, the literature on *structural estimation* provides one way this can be done, essentially by finding a decision structure $[\beta, u, p, D]$ that enables predicted behavior from the DP solution to best approximate behavior we actually observe. This can also be regarded as an "inversion operation" from observed behavior to uncover the underlying structure of the decision problem that is also known as *inverse optimal control* in the engineering literature.[12] However structural estimation has a number of inherent limitations and depends on a key assumption that limits its usefulness for learning the underlying structure, including the curse of dimensionality as discussed in Rust (2014). But even for sufficiently simple problems where the curse of dimensionality is not an issue, a key assumption underlying structural estimation is the standard, "as if" assumption of optimization and unbounded rationality

[12] See, for example, Rust (1994), Eckstein and Wolpin (1989) and Aguirregabiria and Mira (2010) for surveys of this literature.

that underlies most of economics, as I discussed in the introduction. That is, structural estimation learns the underlying structure of a decision problem by *presuming the actors in question have already in effect solved their DPs.* If this is the case, the actor really has no need for the critic, since the actor is already behaving optimally! Another key problem is identification problem: in the absence of fairly strong functional form restrictions, Rust (1994) and Magnac and Thesmar (2002) have shown that dynamic discrete choice models are *non-parametrically unidentified* — that is, there are infinitely many different structures $[\beta,u,p,D]$ that can rationalize observed behavior, i.e. the mapping from from $[\beta,u,p,D] \longrightarrow \delta$ is many to one and thus not invertible without strong additional restrictions such as parametric restrictions on (u,p) (i.e. their functional form is known up to a finite number of unknown parameters). Thus other sources of information, possibly including direct elicitation of preferences and beliefs of the decision maker, may be necessary in order for the critic to be able to learn the structure of the actor's decision problem.

In section 4 I discuss what econometricians might call a *semi-parametric two step learning algorithm* that relaxes the key "as if" optimality and unbounded rationality assumption underlying most of structural estimation that may be applicable for problems where there is sufficient prior information to identify the structure of the DM's decision problem.[13] This approach is most applicable for firms, under the prior assumption that they are expected discounted profit maximizers and have rational expectations. In such cases we can identify $[\beta,u,p,D]$ given sufficient data, and the implied optimal decision rule δ can form the basis for an actor-critic system to advise firms.

Specifically, we can use data from the firm (actor) in question (and other sources), and use non-parametric estimation methods to uncover the actor's existing or *status quo* decision rule, which may not be optimal. Using this estimated decision rule as a *control function* to deal with econometric problems of *endogeneity* that are also commonly present in firm data, it is often possible to obtain consistent econometric estimates of the remaining structure of the firm's decision problem, which include consumer demand and the firm's technological possibilities (e.g. production and cost functions). For publicly traded firms the discount factor can be estimated from stock return data (assuming the capital asset pricing model is a good approximation), or it can be directly elicted from the firm. Thus, there may be enough information for the critic (advisor) to learn enough about the structure of the firm's decision problem to be in a position to solve the firm's problem by DP and provide it with optimal recommended decisions. Since the econometric model has already estimated the (potentially suboptimal) decision rule used by the firm (i.e. its *status quo* decision rule), it is possible to conduct stochastic simulations to compare the optimal and *status quo* policies and quantify the gains from switching to the calculated optimal DP policy. If these computer

[13] See also Dellavigna (2018) who discusses a new literature on structural behavioral economics whose goal is to to relax some of the strong assumptions, including optimization and unbounded rationality, underlying standard structural econometrics.

simulations are sufficiently promising, the actor is more likely to follow the recommended decisions in practice, and further validations of the critic's recommendations can be carried out via *controlled field experiments* as well as before/after comparisons to validate whether the simulated gain in performance from the DP decision rule is borne out in reality.

In situations where the non-parametric two step learning algorithm discussed above can be used to uncover the underlying structure of the actor's decision problem, some of limitations of the supervised learning approach are ameliorated. This algorithn no longer depends on the assumption that we can observe the behavior of an "expert" who is already behaving optimally in order to "train" the DP algorithm. Instead we can train the DP algorithm using potentially suboptimal decisions by a non-expert actor. However the algorithm must be run offline, that is, before it can be used to make recommended decisions we need to gather data from the actor and conduct an econometric analysis to learn the structure of the actor's decision problem and then solve the DP. Thus, this approach contrasts with online or *real time* learning algorithms that we discuss below that can continuously monitor states and decisions made by the actor and iteratively improve their performance concurrent with actual decision making. The real time learning algorithms can potentially adapt to changes in the structure of the actor's decision problem. Offline approaches can be adaptive if they are run in a sort of "batch mode" where datasets on states and decisions observed from the actor are continuously collected and periodically reanalyzed to update the econometric estimates of the underlying structure of the actor's decision problem and recalculate updated DP decision rules based on the latest available data. For this reason it is important not to overemphasize the difference between online and offline approaches to DP.

It is important, however to recognize the problem of *time inconsistency* in the adaptive approach outlined above. Most DPs treat the structure of the decision problem as perfectly known by the decision maker. However if the structure of the decision problem is incompletely known or changes over time in a way that is not explicitly modeled, the approach of solving simpler DPs that treats the structure as fully known, but incorporates some way of sequentially estimating and updating the structure of the problem as time unfolds will generally not result in an optimal, dynamic decision rule for the actual problem that takes the incompletely known or time-varying structure of the decision problem explicitly into account. We can find an optimal time-consistent solution to the problem by solving the problem as a *Bayesian DP* where the DM has a prior distribution over parameters of the objective function or in the law of motion for the state variables that the DM does not know or which could evolve over time. The prior distribution for these parameters are then updated in real time as the system is controlled, using Bayes Rule.

However in the absence of cases where the prior and posterior distributions are members of simple conjugate prior families, the resulting Bayesian DP problem will generally have an infinite dimensional state variable, i.e. the current posterior distribution over the unknown parameters that the DM is trying to learn.[14] It is generally computationally infeasible to solve DPs with infinite dimensional state variables, so most applications of DP involve the time-inconsistent adaptive "learning" approach discussed above, and thus are not optimally resolving the tradeoff between experimentation and exploitation which is captured in Bayesian DP problems such as the multi-armed bandit problem noted above.[15]

In this section I use the term "supervised learning" to refer to high level of human intervention required to carry out the offline approach to solving DPs discussed above but it should not be confused with "supervised learning" as it is used in the literature on ML, where it refers to training an algorithm to extrapolate and generalize observations of decisions of an expert, who is typically a human being. This latter type of learning would be akin to non-parametric regression to uncover an optimal decision rule $d = \delta(s)$ from a data set of pairs (d_i, s_i) of decisions and states of an expert human DM who might be considered to be a "role model" for some other DM who is not following an optimal policy. If the structure of the decision problem is the same for the role model DM and the DM in question, then this simpler type of supervised ML might be a reasonable thing to do, and completely avoids the need to actually solve the DP problem (since the role model is assumed to have already done it).

A deeper form of supervised learning (that includes actually solving the DP problem) is required if a) we do not believe there is another DM that is behaving optimally and which can serve as a role model, or b) the structure of the decision problem of the DM in question is different from the structure of the decision problem faced by any potential role model we might consider. In such cases, "supervised learning" refers to both the need to gather data and use econometric methods to unover the structure of the DM's decision problem, as well as to explicitly solve the DP problem to provide recommended decisions to the particular DM in question. Both of these tasks take place offline and require a high degree of human intervention to conduct the econometric analysis of the data and the solution of the DP problem.

In the next section I discuss DP algorithms from the literature on RL that can operate in real time, in an

[14] See Kumar (1989) for a survey on adaptive control that includes non-Bayesian adaptive algorithms such as the use of recursive maximum likelihood or least squares methods to learn the unknown parameters of the problem during the process of control, which are more tractable that full Bayesian DP, but requires repeated re-solution of the DP problem as the parameter estimates change over time. The focus of this latter work is on establishing on whether the unknown parameters θ^* can be consistently estimated as $T \to \infty$, resulting in an adaptive and asymptotically optimal decision rule (i.e. a decision rule that is optimal when the decision problem is fully known, including the parameter θ^*.

[15] There is still another meaning of the term "learning" in dynamic decision problems that may or may not directly be related to Bayesian updating of unknown quantities. Examples including "learning by doing" such as when a customer tries a drug or a new consumer product and learns (often after a single consumption experience) whether the drug has side effects or the product is one the consumer likes or not. These types of learning effects play an important role in pharmaceutical demand and can be important drivers of R&D expenditures by drug companies. See Crawford and Shum (2005), Chan and Hamilton (2006) and Hamilton, Hincapie, Miller, and Papageorge (2018).

unsupervised manner, and are model-free, and thus achieve the ideal of ML methods by avoiding human intervention to program and update the algorithm, or estimate econometric models to learn the structure of the DM's decision problem. The *Alpha Zero* algorithm is an example of this type, and it attained superhuman performance in Chess and Go after being trained in an unsupervised manner, entirely through self-play.

2.2 Real time solution of DPs va reinforcement learning

The DP literature proved highly influential to researchers in AI because "DP provides the appropriate basis for compiling planning results into reactive strategies for real-time control, as well as for learning such strategies when the system being controlled is incompletely known." (Barto, Bradtke, and Singh (1995), BBS, p. 82). The value iteration and policy iteration algorithms for offline solution of DPs inspired AI researchers interested in RL to develop stochastic versions of these algorithms for solving DPs that can converge asymptotically to optimal decision rules and value functions even for systems that are being controlled by these algorithms in real time and without the requirement for an explicit model of the structure of the DM's decision problem. Examples of RL algorithms include *temporal difference learning* Sutton (1988) and *Q-learning* Watkins (1989), as well as the Real time DP (RTDP) algorithm proposed by Barto et al. (1995). Not all versions of these algorithms are model free, and not all of them are run only in real time. In fact, many of them are trained offline and to do this, the analyst needs to have some type of model of the underlying structure of the problem to be able to *simulate* transitions and payoffs. However after initial training, most of these algorithms can operate in real time and their performance tends to improve over time with experience, so they qualify as learning algorithms that improve as a result of *unsupervised learning* in the sense that "DP-based learning algorithms are examples of reinforcement learning methods by which autonomous agents can improve skills in environments that do not contain explicit teachers." (BBS, p. 82).

The original work focused on establishing the convergence of these algorithms in finite state stationary MDPs. This literature has developed in various directions and is now known by several terms including *neurodynamic programming* (ND) Bertsekas and Tsitsiklis (1996) and *approximate dynamic programming* (ADP) Powell (2010) where neural networks are used instead of table-lookups of values for problems with continuous state and decision variables or finite MDPs such as Go where there is a huge number of possible states and decisions. For example Mnih et al. (2015) showed that by combining Q-learning with multi-layer or "deep" neural networks, the resulting *Deep Q-Networks* "can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning" and "receiving only the pixels and the game score as inputs, was able to surpass the performance of all previous algorithms and achieve a level comparable to that of a professional human games tester across a set of 49 games, using the same algorithm, network architecture and hyperparameters." (p. 529).

BBS proved the convergence of RTDP using previous work on *asynchronous DP algorithms* by Bertsekas (1982) and Bertsekas and Tsitsiklis (1989) which was developed to prove the convergence of the traditional value function iteration and policy iteration algorithms in parallel computing environments where not all processors are equally fast or perfectly synchronized. This implies that values for some states are updated or *backed up* (in the terminology of BBS) at different times, yet as long as these update or backups of the value function at all possible states occur infinitely often, the sequence of value functions and decision rules $\{V_k, \delta_k\}$ produced by asynchronous DP algorithms still converge to to the optimal values given by the Bellman equation for stationary infinite horizon MDPs, (2) and (3). Recall that the standard value function iteration can be written as $V_{k+1} = \Gamma(V_k)$ where the optimization problem to determine the optimal decision rule $d = \delta(s)$ is done for *all* states s, that is the current value $V_k(s)$ is updated and "backed up" as $V_{k+1}(s)$ for *all states s*. RTDP is simply a natural extension of the idea of asynchronous DP to a case where only *the currently occupied state s_k at time k is backed up, but the values for other states that did not occur are left alone.* That is, RTDP implies an updating rule for V_{k+1} of the form

$$V_{k+1}(s) = \begin{cases} \max_{d \in D(s)} \left[u(s,d) + \beta \int V_k(s')p(s'|s,d)\right] & \text{if } s = s_k \\ V_k(s) & \text{if } s \neq s_k \end{cases} \tag{5}$$

where s_k is the realized state of the decision process at time $t = k$, which is also the index for the iterative updating of the value function and we let $\delta_{k+1}(s)$ be the corresponding updated decision rule, the value of d that solves the maximization problem in (5) at state s_k. Then, the DM takes the decision $\delta_{k+1}(s_k)$ resulting in a new realized state $\tilde{s}_{k+1}$ d at time $t = k+1$, a draw from the transition probability $p(s'|s_k, \delta_{k+1}(s_k))$, and the process continues on this way forever. Because of the stochastic nature of the updates, the sequence of value functions and decision rules from RTDP, $\{V_k, \delta_k\}$, is also stochastic. Under certain conditions, this sequence will converge with probability 1 to the true solution (V, δ), generalizing the standard convergence theorem on successive approximations for contraction mappings.

A key assumption is that *all states must be visited infinitely often* but it can be difficult to provide high level assumptions to guarantee this happens in real time, especially if the process has transient states that are visited infrequently. Thus, in practice, Barto et al. (1995) discuss *trial based RTDP* which is "RTDP used with trials initiated so that every state will, with probability one, be a start state infinitely often in an infinite series of trials." However these extra trials can only be done offline and so the algorithm can no longer be described as "real time." Further, note that neither variant of RTDP is "model free" since the updating formula (5) requires knowledge of the problem structure $[\beta, u, p, D]$ and requires explicit optimization and numerical integration.

Q-learning is a true real-time and model-free learning algorithm, and is one of the reasons it has been used in the above cited work by DeepMind to train algorithms to achieve human level and even superhuman levels of skill in a variety of two player games such as chess and Go. Watkins (1989) defined the $Q(s,d)$ function as the right hand side of the Bellman equation, i.e.

$$Q(s,d) = u(s,d) + \beta \int V(s')p(s'|s,d) \tag{6}$$

Economists refer to $Q(s,d)$ as the *decision-specific value function* and it plays an important role in the literature on structural estimation of dynamic discrete choice models since knowledge of the Q function is sufficient to describe the choices of a decision maker. Specifically we have $V(s) = \max_{d \in D(s)} Q(s,d)$ and $\delta(s) = argmax_{d \in D(s)} Q(s,d)$ and Q itself can be regarded as the unique fixed point of a contraction mapping, $Q = \Lambda(Q)$ given by

$$Q(s,d) = \Lambda(Q)(s,d) \equiv u(s,d) + \beta \int [\max_{d' \in D(s')} Q(s',d')]p(s'|s,d). \tag{7}$$

Let Q_k denote the estimate of the Q function at time $t = k$ when the current state is s_k. The DM is assumed to act in what computer scientists call a *greedy* manner, i.e. by taking a decision $d_k = argmax_{d \in D(s_k)} Q_k(s_k,d)$ that has the highest value. Let $\tilde{s}_{k+1}$ be the new state resulting from this choice, i.e. it is a draw from the transition probability $p(s'|s_k,d_k)$. Given $(\tilde{s}_{k+1}, s_k, d_k)$ the DM updates Q_k as follows

$$Q_{k+1}(s,d) = \begin{cases} Q_k(s_k,d_k) + \alpha_k \left[u(s_k,d_k) + \beta \max_{d' \in D(\tilde{s}_{k+1})} Q_k(\tilde{s}_{k+1},d') - Q_k(s_k,d_k)\right] & \text{if } (s,d) = (s_k,d_k) \\ Q_k(s,d) & \text{otherwise} \end{cases} \tag{8}$$

where $\alpha_k > 0$ is a stepsize parameter that should decrease to zero as $k \to \infty$ but not too fast, as I discuss below. Thus, the updating rule in Q-learning is similar the RTDP updating rule (5): it only updates Q_{k+1} at the state, decision pair (s_k,d_k) that actually happens at $t = k$, and it is stochastic because it involves the stochastic successor state $\tilde{s}_{k+1}$ that results from the decision d_k in state s_k. However unlike RTDP, Q-learning is indeed *model-free* since the updating rule (8) only depends on the *realized payoff* $u(s_k,d_k)$, but does not require an explicit functional form for the DM's reward function $u(s,d)$ or the transition probability $p(s'|s,d)$. Tsitsiklis (1995) proved that Q-learning converges with probability 1 to the true Q function, the unique fixed point to (7) in finite MDPs. He showed that the iteration (8) constitutes a form of *asychronous stochastic approximation* for finding the fixed point of Λ.[16] The key assumptions underlying

[16] The adjustment term inside the brackets and multiplied by α_k in equation (8) is stochastic but has conditional mean zero when $Q_k = Q$, the true fixed point, since from (7) we see that $E\{u(s,d) + \beta \max_{d' \in D(s')} Q(s',d') - Q(s,d)|s,d\} = 0$. Thus, the term in the brackets multiplied by the stepsize α_k in (8) a deterministic part, $\Lambda(Q_k)(s,d) - Q_k(s,d)$ plus another term that constitutes random "noise." Since Λ is a contraction mapping, the deterministic part tends to zero as $k \to \infty$ and the stochastic noise tends to zero due to the decreasing step sizes α_k. The main difference of this algorithm and the well known method of stochastic

the asynchronous stochastic approximation algorithm are that 1) Λ is a contraction mapping with a unique fixed point Q, 2) each possible state/decision pair (s,d) is visited, and thus updated, infinitely often, and 3) α_k obeys the standard rate conditions for stochastic approximation, i.e. $\sum_{k=0}^{\infty}\alpha_k = \infty$ and $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$.[17]

Q-learning suffers from the same problem as RDTP, namely it can be hard to establish conditions that guarantee that each point (s,d) is visited/updated infinitely often as $k \to \infty$. One way to achieve this is to impose a degree of random experimentation on the DM, such as by adding stochastic shocks $\varepsilon(d)$ to the calculated Q_k values, so decisions are given by the value of d_k that maximizes $Q_k(s_k,d) + \sigma_k\varepsilon(d)$ where σ_k is a sequence that converges to zero at an appropriate rate. These random shocks that affect the DM's choice of decision coupled with sufficient ergodcity in $p(s'|s,d)$ can lead to sufficient random experimentation to guarantee that every (s,d) point is updated infinitely often, and thus ensure the probability 1 convergence of the stochastic sequence $\{Q_k\}$ to the true value Q.

Related methods such as temporal difference learning are recursive stochastic algorithms to calculate V_δ, the policy implied by a decision rule δ, either in real time or via stochastic simulations. Thus TD is an approximate way to do the policy valuation step of policy iteration, avoiding direct solution of a linear system that becomes infeasible when there are huge numbers of states. Bertsekas and Tsitsiklis (1996) relate policy iteration to the actor-critic algorithms studied in the RL literature: the "policy evaluation step is viewed as the work of a *critic,* who evaluates the current policy, i.e. calculates an estimate of V_{δ_k}, the value of a policy δ_k. The policy improvement step is viewed as the work of an *actor,* who tskes into account the latest evaluation of the critic, i.e. the estimate of V_{δ_k}, and acts out the improved policy δ_{k+1}." (p. 36).

The stochastic convergence results discussed above are only for finite MDPs and the algorithms only work well for problems with a relatively small number of states and decisions. Otherwise if there are too many s or (s,d) pairs that are not backed up, the algorithms may fail to discover optimal strategies especially if initialized from V or Q values that are far from optimal.[18] Thus, for relatively small problems where the structure of the decision problem $[\beta,u,p,D]$ is known, the standard value function iteration and policy iteration methods described in the previous section are generally faster and produce far more accurate approximate solutions. However for moderate to large scale MDPs the curse of dimensionality starts to become evident, and it becomes increasingly difficult to employ the standard types of algorithms and guarantee a sufficiently precise solution to the problem. For sufficiently large scale problems with

approximation Robbins and Munro (1951) for finding zeros or fixed points of functions defined by conditional expectations is the asynchronous nature of the updates: only the current state/decision component (s_k,d_k) of Q_k is updated at iteration k instead of at all possible values under standard stochastic approximation.

[17] If the step sizes are random variables they must be positive and obey these conditions with probability 1.

[18] There are cases where values of V or Q that are not close to being fixed points of their respective operators can nevertheless imply decision rules that are close to optimal, see for example, the discussion in chapter 6 of Puterman (2005).

multiple continuous state variables the stochastic learning algorithms start to be competitive for the simple reason that the traditional types of non-stochastic algorithms are generally no longer even computationally feasible. However the convergence results for Q-learning and other types of RL are no longer applicable either. Even for finite MDPs such as board games such as chess or Go, the number of possible states and decisions becomes far too large for the standard RL algorithms based on "table lookup" of values to be effective, so some sort of interpolation/extrapolation of values becomes necessary in order to implement these algorithms.

Just as the standard numerical algorithms discussed in the previous section have relied on parametric approximations to V or δ using a relatively small number of unknown parameters ω (such as approximating V via various linear combinations of basis functions or via neural networks, etc), these same sorts of parametric approximation have been applied in the RL liteature but where the parameters ω are sequentially updated via various types of iterative stochastic gradient methods that avoid direct solution of the least squares approximation problem (4). Bertsekas and Tsitsiklis (1996) and Powell (2010) provide a comprehensive discussion of the various ways that RL algorithms such as TD, RTDP and Q-learning can be combined with various types of function approximation methods, but they caution that "convergence to [V or Q] cannot be expected, for the simple reason that [V or Q] may not be within the set of functions that can be represented exactly within the chosen architecture." (p. 256).

Neural networks have proved attractive as an approximating class of functions due to their flexibility and the fact that they can approximate well a wide range of functions using relatively few parameters, see Barron (1994). However the Achilles heel of neural networks is that the implied least squares problem (4) typically has a profusion of local optima, making it very computationally intensive to search for parameter values ω that globally minimize the value function residuals, which is a necessary (though not sufficient) condition to find a good approximation to the true V or Q function. As a result, a great deal of "art" (i.e. practical experience, and perhaps luck) is required to design a neural network architecture (i.e. choose the number of hidden layers and specify where and how the different state variables enter these layers) and to train it so that the outcome is satisfactory. Thus, neural networks should not be regarded as a pancea or cure for the curse of dimensionality: they merely deal with one form of the curse of dimensionality (i.e. approximation of functions of d variables using a vector of parameters ω that increases only linearly in d) at the expense of another (the curse of dimensionality involved in globally minimizing the least squares criterion in ω). It is quite easy for training to find a local but not global minimum, and a poorly trained neural network approximation can perform poorly. For example a neural network trader peformed poorly in the double auction tournament run at the Santa Fe Institute Miller, Palmer, and Rust (1993).

However there have been significant advances in the literature on RL and training of deep neural networks, as evidenced by the impressive human and superhuman level of play acheived in a wide range of games acheived by *Deepmind's* deep-Q networks cited above that demoinstrate "that a general-purpose reinforcement learning algorithm can achieve, *tabula rasa,* superhuman performance across many challenging domains." (Silver (2017), p. 2). The fact that the *AlphaZero* was trained from scratch and entirely via self-play and was able to beat the leading competing chess program *Stockfish* after just 4 (wall clock) hours attracted considerable publicity, but consideration should be given to the huge number of processors required to do this and the huge number (over 300,000) of games that must be played to train the algorithm to the point where it is capable of a high level of performance.

As I discuss below, I am not aware of other real world applications where RL (combined with deep nets, ordinary nets, or other approximation architectures) has achieved comparable success. I think this reflects a huge advantage associated with the use of RL to train algorithms to play well in ordinary board games: *the structure of the decision problem is mostly known and the problem can be easily simulated.* The payoffs of the players in a game such as chess is quite simple: say, $+1$ for a win, -1 for a loss and 0 for a draw. Further, the rules of chess and other board games such as Go are fully known and can be rapidly simulated on a computer, making it possible to play out millions of moves in offline training sessions on thousands of processors in a relatively small amount of wall clock time. The main aspect of the game that is not known (and must be learned) is the response probability of the opponent. However the training pits a copy of the algorithm to play against another copy of itself, so and this facilitates its ability to learn the probability distribution over the moves of its opponent in different states.[19]

It is less clear that real time DP and RL algorithms are appropriate for approximating DP solutions in the much fuzzier and more complex real world environments that consumers and firms ordinarily operate in. For these problems the structure of the decision problem is not so easily specified and for firms, may include the need to learn about consumer preferences and demand including how demand is affected by decisions of other competitors and business cycle fluctuations. Though the idea of "model free learning" is appealing, it seems unlikely that individuals or firms would have the willingness to follow recommendations of a poorly trained and initialized deep Q-network if its recommended actions do not seem sensible.

[19] Chess is an example of a *dynamic directional game* (see Iskhakov, Rust, and Schjerning (2015) for a definition) and may have multiple equilibria. It is not clear that, despite the superhuman performance of *AlphaZero*, the training converges to Nash equilibrium strategies, but could cycle as in the well known "fictitious play" algorithm for finding Nash equilibria, Brown (1951). Iskhakov, Rust, and Schjerning (2018), using the *recursive lexicographical search* (RLS) algorithm of Iskhakov et al. (2015), characterized all equilibria to a class of DDGs with Bertrand price competition and leapfrogging investments that have many fewer states than chess, yet have billions of equilibria. It is not clear that RL algorithms trained on copies of themselves will also result in strategies that are approximate best responses to other strategies, and it may be possible to train new generations of algorithm to exploit weaknesses in *AlphaZero* similar to what was done in the "evolutionary tournament" of competing trading strategies in the double auction market, as reported in Miller et al. (1993).

Unless real decision makers follow the recommendations of a RL algorithm, it is unclear how it can be trained to improve its performance over time. Most real world decision makers do not make enough decisions to provide the thousands to millions of training instances needed by RL algorithms to learn in real time, and few have the patience to train them in a simulated decision making environment.

Thus, it is clear that if RL is to be successful it must have a capability to either a) aggregate experience learned from advising "similar" decision makers, or b) conduct offline simulations to train itself to be more effective form the much smaller number of real world opportunities where it can advise individuals or firms on actual decisions. However it is not clear how ML approaches by themselves can recognize when individuals or firms are sufficiently similar in the specific sense that the structure of their decision problems are nearly the same, that a recommended decisions that represents good advice for one will be equally good for others, and the collective data and training experience can be pooled and shared. However it may be possible to effectively aggregate experience and knowledge across different individuals, as examples from something as seemingly subjective as choice of clothing fashions suggest.[20]

However it is less clear if ML methods can be effective for more challenging problems such as life cycle decision making that I discuss in the next section. The structure of an individual's life cycle problem is far more complex and includes subjective beliefs about many uncertain quantities, such as health, preference for work vs leisure and for different types of goods/services, the prices of goods/services and how they vary over time, rates of return on different investments, beliefs about current and future job opportunities, etc. etc. Given all these dimensions, it is unclear how to aggregate different individuals into relatively homogeneous subsets who have similar preferences and beliefs.

Further the structure of most decision problems is sufficiently complex that it seems unlikely that it can be learned in a completely non-parametric, model-free manner by RL algorithms such as deep Q-networks. Training these networks to solve reasonably complex real world problems can probably only be done offline using simulations. But if the structure of the decision problem is unknown, how do we construct a realistic simulation environment to train the algorithm? This is the catch-22 and Achilles heel involved in trying to apply RL algorithms to solve real world problems. Human intuition and understanding seem necessary in order to build an adequate simulation model that provides a reasonable approximation to the action real world decision making environment that is needed to solve/train/test the DP algorithm offline in order to have any assurance that it will perform well in an on-line situation in real time.

[20] For example the company Stitch Fix uses ML algorithms to recommend new clothing designs to its customers and learns from repeated feedback from individual customers, but also across customers with apparent similar tastes. As its CEO notes, "As time goes by, algorithms learn from actual customers, both individual and in aggregate, how to think about clothes. This is possible thanks to the feedback data collected from customers, which is transmitted back to the algorithms so that they can see how their decisions worked in real life – and use this information to constantly improve their decision-making formulas (machine learning)." (Forbes (2018)).

Thus, it seems that both of the approaches that I have outlined in this section for solving DPs depend critically on the ability to *learn the structure of the decision problem.* To my knowledge ML is not able to acquire this type of knowledge from scratch: it is related to a fundamental and unsolved puzzle of *abduction* i.e. the process humans use to form *models* that provide efficient though imperfect representations of the world that may be key to human intelligence.[21] Without good knowledge of the structure of the decision problem, it is not clear that DP algorithms of any type will be useful to real world decision makers if they fail to grasp the actual problem and the constraints, opportunities, and challenges that real world decision actually confront.

3 Applications of DP to improve individual decision making

This section is relatively short because I am not aware of many applications where DP has been used to help individuals improve their decision making. I start by discussing the *only* example that I am aware of where DP is definitely being used to improve individual decision making. I then discuss a number of other examples where some form of DP is probably being used as a decision assist tool. Next I discuss possible reasons why DP has not been applied to help individual decision making so far. I discuss the substantial empirical evidence of suboptimality in individual decision making, and conclude that there ought to be a wide range of applications where DP can help individuals make better decisions, though we lack tools to evaluate whether DP actually improves individual welfare.

3.1 DP for improving life cycle decision making

Laurence Kotlikoff is a leading economist who has made seminal contributions to public economics and our understanding of the life cycle model, including personal financial decisions and retirement. He notes that "In the course of these studies, I realized two things. First, economics' approach to financial planning is very different from the conventional approach: it generates quite different and much more sensible advice. Second, economists have an obligation to use their tools and science to help people make proper life-cycle financial decisions." Kotlikoff founded Economic Security Planning, Inc in 1993 to operationalize his vision by creating the first "economics-based personal financial planning software." The company now has over 100,000 customers using its *MaxFi* software which "uses iterative dynamic programming methods developed by our founder". Kotlikoff confirms that DP is absolutely the key to the quality of the advice his company provides its customers, and notes "since all questions of personal finance begin and end with a household's spending path, no one can answer any personal finance question

[21] See Heckman and Singer (2017) and Griffiths and Tenenbaum (2009).

of any kind without using dynamic programming. That's why I humbly submit that we have the world's only real financial planning tool." The reason why a DP-based approach is potentially so superior to other standard approaches to financial planning is that DP results in *state-contingent consumption/savings plans* that account for dynamically changing budget constraints in a realistic fashion. Standard approaches to financial planning typically rely on a fixed *desired consumption plan* that fails to to account for the future consequences of current decisions, and how consumption should respond to contingencies such as divorce, unemployment, unexpected health problems and so forth.

For proprietary reasons, Kotlikoff will not disclose the details of the DP algorithm his company developed, but in terms of the two different classes of DP algorithms discussed in section 2, *Maxfi* falls in the class of standard, human-coded numerical solution method and is not a real-time ML/RL algorithm discussed in section 2.2. The company has a patent on its own specially developed algorithm for solving the customer's life-cycle problem extremely rapidly using a multi-stage, multi-pass method with a "special technique to ensure our DP is precise to many decimal places". Kotlikoff notes that "We haven't done full dynamic stochastic programming yet, but are heading down that path at least for research purposes." Skinner (2007) illustrates calculated target non-housing wealth calculated from a forerunner to *Maxfi* called ESPlanner and remarks, "In using ESPlanner, I was struck by how many factors — far more than just the standard economic variables — had enormous effects on target wealth. Even a simple spreadsheet program can engender that critical wake-up call to think more about planning for retirement." (p. 76).

How does *MaxFi* learn about the structure of their customers' decision problems, i.e. their preferences (including preferences for work versus leisure), and the law of motion for the relevant state variables that include expectations of future earnings, job loss, job opportunities, health problems and mortality? Maxfi uses elictation of preferences via questions to a number of factual and hypothetical questions including inferring mortality risk by specifying the max age of life as the planning horizon and preferences for work versus leisure via questions on desired retirement ages and their preferred age-consumption profile. *MaxFi* then replicates that profile as closely as possible subject to the cash flow constraints. The program also provides clients with a simulation laboratory where they can explore the consequences of different assumptions and future contingencies. For example, users can "specify cuts or increases in each type of tax they face and also in SS benefits. We also let them override our default assumption about the growth rate of Medicare Part B premiums. Otherwise, we encourage users to do lots of what ifs." Once Maxfi has solved the customer's life cycle optimization problem it communicates the solution by means of life cycle simulations that "show the living standard impact of important financial and life cycle decisions like switching jobs, retiring early (or later), postponing Social Security, and much more."

Kotlikoff does not have direct information on how many of his company's customers actually follow the *MaxFi's* recommendations. He is open to allowing independent researchers to evaluate the quality of the advice and its impact on its customers' welfare and he notes that "The Atlanta Fed has licensed a research version of our code, so we're working with them on this project." There are interesting questions on how to design an experiment to measure the welfare impact of the *MaxFi* software's recommendations. Presumably it would require an experiment lasting many years, following a "treatment group" who has access to MaxFi and a "control group" that does not. However it is not entirely clear how to measure and compare the welfare of the individuals in the control and treatment groups though it seems possible to compare more objective indicators of welfare such as net worth profiles, level of work and earnings and leisure time as well as subjective indicators of overall happiness.

Absent a more scientific evaluation, the company website posts a number of customer testmonials of the value of the software. For example Bob, a retired mathematician, says "I find MaxiFi Planner to be extremely helpful. It enables me to input a great level of detail regarding our current situation, and is then easy to use in generating very valuable and insightful future scenarios." Gil Weinreich of the financial website *SeekingAlpha.com* writes "MaxiFi Planner is a simple and elegant solution for financial advisors seeking to advise clients or for unadvised investors looking to optimize their own plan." Evidently at least 100,000 think that recommendations from *MaxFi* are worth the $99 annual subscription fee, so this provides a lower bound on the willingness to pay and thus value-added from this software.

There have been academic studies of the question of how closely individuals follow predicted savings/consumption paths from the life cycle model. Scholz, Sheshadri, and Khitatrakun (2006) use DP to "solve each households optimal saving decisions using a life cycle model that incorporates uncertain lifetimes, uninsurable earnings and medical expenses, progressive taxation, government transfers, and pension and social security benefits. With optimal decision rules, we compare, household by household, wealth predictions from the life cycle model using a nationally representative sample." (p. 607). They conclude that the individuals they analzed "are saving enough to maintain living standards in retirement. And second, the life cycle model provides a very good representation of behavior related to the accumulation of retirement wealth." In fact, they find that "the life cycle model does a much better job of matching the cross-sectional distribution of wealth in 1992 than a naive model in which households save an income- and age-varying fraction of income." However the authors note that their favorable findings pertain to a particular generation born between 1931 and 1941 (a generation whose saving behavior may have been imprinted from their parents' hardships in the Great Depression) and they caution that "we need to be careful in generalizing our results for the HRS cohorts to younger households." (p. 638).

For the baby boom generation there is evidence of financial illiteracy and widespread undersaving and lack of planning for retirement. The survey by Lusari and Mitchell (2014) exposes a shocking degree of financial illiteracy, even on basic questions such as simple interest rate calculations or knowledge of the benefits of diversification. A study by Institute (2018) (an association that represented professional retirement planners) found that an astounding 42% of boomers have *no* retirement savings, and 70% have less than $5000 in liquid funds for emergencies. Among boomers who have saved for retirement 38% have less than $100,000 and only 38% have calculated the amount they will need to retire. In order to attain their reported desired retirement consumption levels, nearly half of boomers will need to increase their annual retirement income by about $45,000, which would require additional retirement savings of about $430,000. Similar warnings about inadequate retirement savings appear in academic studies such as Ellis, Munnell, and Eschtruth (2014) and the references they cite. Reductions in traditional pension savings (including 401(K) plans) has not been offset by corresponding increases in personal saving such as IRAs. As a result, a large fraction of boomers will end up depending on Social Security for most of their retirement income. For many of these individuals the most likely scenarios are either a) the lack of retirement savings will force dramatic cuts in retirement consumption, or b) they will have to work longer. Data from the Bureau of Labor Statistics indicates that the secular trend in early retirement ended about three decades ago and has reversed since then, and currently about 20% of Americans over 65 continue to work. Of course, either a) or b) could be consistent with an optimal life-cycle decision rule, especially in light of improvements in health and longevity and higher demand for older workers. However all three of these are in question for many middle to lower income individuals who lack good education and job skills that are in demand in the modern economy. The work of Case and Deaton (2015) shows *increasing mortality* for many in this category, due largely to obesity, alcoholism and opoid addiction. Stagnating real wages and de-industrialization and the effect of automation on displacing low-skilled workers makes the working longer strategy appear to be a rather grim scenario for many of these boomers.

Laboratory experiments designed to mimic the life cycle problem have also provided mixed evidence on individuals' ability to solve the life cycle problem and plan and save optimally for retirement. Johnson, Kotlikoff, and Samuelson (2001) present the results of "an experimental study in which subjects were asked to make preferred life cycle choices under hypothetical life cycle economic conditions. The questions in the experiment are designed to test the model's assumption of rational choice and to elicit information about preferences. The subjects' responses suggest a widespread inability to make cohorent and consistent consumption decisions." They find subjects make substantial, systematic errors in decision making that "strongly reject the standard homothetic life cycle model." (p. 1). A study by Carbone and

Duffy (2014) focuses on a deterministic life cycle model but finds that "subjects have difficulty solving a 25-period lifecycle consumption optimization problem despite the absence of any uncertainty and given two opportunities to go about it" and "tend to save too little relative to the optimal path (i.e., they over-consume), a phenomenon that has been found in other studies as well." (p. 427). Other experiments conducted recently at the Economic Science Institute at Chapman University expand the life cycle model to include investments in "health capital" with a range of fairly realistic modeling scenarios that are solved by DP. James and Tracy (2018) find that subjects' behavior shares many qualitative features of the DP solution but does not precisely conform to the predictions, resulting in outcomes that are between "69% to 77% efficient depending on the treatment arm" where "efficiency" represents the ratio of the subject utility to the utility a DP strategy would obtain facing the same shocks and initial conditions. They conclude that "Overall, they did pretty well in individual periods but the inefficiency compounds over the 32 periods".

It is probably impossible to provide any definitive conclusion about whether the individuals are or are not capable of solving the life cycle problem, but it does seems clear that there is significant evidence that many individuals have difficulty, and may benefit from advice of life cycle planning models such as the *MaxFi* software. What remains unclear from a methodological standpoint, however, is a) how best to uncover or elicit an individual's preferences and beliefs? and 2) how to evaluate the welfare impact of being exposed to recommended decisions and advice from a formal DP algorithm?

3.2 DP for improving consumer search decisions

As I noted above, there seem to be many other areas where DP could be used to improve individual decision making and welfare. One area of rapid growth is in *search robots* that search for the lowest price for an individual who knows they want to buy a certain relatively standardized item or service (e.g. a specific make/model of car, or a plane flight to a specific destination and departure/return dates). The search problem can be formulated as particular type of DP known as an *optimal stopping problem* and the optimal strategy typically takes the form of a reservation price rule: stop searching and buy at the first price quote below a calculated reservation price. Individuals appear to search in a suboptimal fashion: "Experimental evidence, however, suggests that actual stopping decisions are characterized by a distribution of stopping values. For example there is often a tendency to stop too soon when confronted with a descending series of observed offers and stop too late when faced with an ascending series" Hutchinson and Meyer (1994) p. 377. Once a consumer decides to purchase an item or service, it clearly improves their welfare if they can buy at lowest possible price. Thus optimal price search is an example of the principle of decomposition where it seems possible that DP-based algorithms could be highly effective and valuable. Optimal stopping problems are relatively easy to solve, but unfortunately I cannot confirm specific commercial consumer search services solve stopping problems using DP. However several of these services could be doing something close to DP.

One such service is *Farecast* a company founded by Oren Etzione, a professor of computer science at the University of Washington, who is now CEO of the Allen Institute for Artificial Intelligence. "A poster child of the Big Data revolution, Farecast analyzed hundreds of billions of airline ticket prices using a machine-learning algorithm and told consumers when to buy. The acquisition helped make price prediction the key differentiator of Bing Travel, a core asset of Microsofts new 'decision engine."' Fung (2014). Though Microsoft subsequently removed Farecast from Bing, the Kayak travel website created its own airline fare search engine that follows specific flights and advises consumers on the best time to buy. In the next section I discuss examples showing the effectivness of DP-based algorithms for the optimal timing of purchases of inventories by businesses. Though I cannot be sure Farecast's or Kayak's algorithms are based on similar ideas, Fung (2014) notes that "The Kayak algorithm is amazing at capturing value when the opportunity presents itself" and he provides an example of a price trajectory from Los Angeles to Chicago where Kayak "recommended purchasing on the second day, right before the fare overshot the 14-day-out price and never came back. This is a sure sign of intelligence."

3.3 DP for improving performance in games

I am disappointed not to be able to offer many more examples where DP has helped improve individual decision making. As I noted in the introduction, there are three possible explanations for why DP has not been widely adopted so far. The first of these explanations is that individuals are very intelligent and already act "as if" they had solved a DP problem for an optimal strategy, if not via the standard algorithms described in section 2.1 then via experience and trial and error, perhaps using a biological version of a RL algorithm described in section 2.2 using the incredibly powerful human neural network. Besides the mixed evidence presented above on individuals' ability to solve their own life cycle problems and the evidence on suboptimal search behavior presented above, I find the fact that the world's best chess and Go players are no longer human beings, but deep Q networks trained by Google's DeepMind subsidiary, provides convincing evidence against the unbounded rationality hypothesis that Simon criticized. Unfortunately while this represents an amazing success for DeepMind and AI as a whole, their algorithms may prove to have little value for human chess or Go-players: what fun is it to play an opponent you can never beat and what good is their advice if such advice is prohibited from being used in tournaments involving only human players?

It is possible that the superhuman chess and Go algorithms can help, ironically, to train *humans* to play these games better, perhaps by improving their ability to look ahead and conform more closely to the backward induction logic of DP. There is a fairly extensive literature on laboratory experiments that reveals the limitations in human strategic thinking and planning in game contexts. For example Camerer (2003) discusses numerous experiments that convincingly demonstrate that human subjects do not use backward induction to make decisions. For example he describes a "mouselab" alternating offer bargaining experiment where human subjects play against computers programmed to play equilibrium strategies. The human subjects would have to click on boxes in the screen to see payoffs at different stages in the bargaining game, information that was necessary for them to form best-replies to their computerized opponent. Camerer notes that "The fact that they don't always open the future boxes is a death blow to the strong form of the backward induction hypothesis." (p. 422).

This conclusion is also confirmed in other experimental studies of dynamic decision making in non-game contexts. "Specifically, although the normative solution of finite horizon dynamic decision problems is generally achieved through backward induction … few individuals actually use such a process, preferring instead to engage in forward induction over an extremely limited horizon (one or two periods)" Hutchinson and Meyer (1994) p. 373.

Of course the fact that human subjects are not *literally* using backward induction in their reasoning in playing dynamic games does not automatically mean they must be playing suboptimally or stupidly. The RL algorithms described in section 2.2 use repeated simulated play that can be viewed as a form of forward induction that helps train the algorithms and enables them to improve their performance over time. Camerer provides examples of experiments where similarly, via experience and learning, human subjects are able to conform more closely to equilibria consistent with backward induction reasoning in some cases: "after training in backward induction, they do make equilibrium offers to computerized opponents (so they are cognitively capable of backward induction, they just don't do it instinctively)." (p. 430). Further, he discusses various formal models of human learning in games (including RL) and concludes "there is no doubt that simple learning rules can approximate learning better than equilibrium (which assumes none)" (p. 782).

3.4 Discussion

Years ago I wrote a paper "Do People Behave According to Bellman's Principle of Optimality?" Rust (1992) and my answer to this question at that time was, "in general, no." The evidence I have seen since I wrote that paper continues to confirm and reinforce my earlier conclusion. However the present article can be viewed as asking a different question: "Should people behave according to Bellman's principle of optimality?". This is a normative question and the answer depends on the extent to which we believe that computer algorithms that attempt to solve a DP approximation to decision problems individuals wish to solve more effectively in their daily lives will actually be able to do a better job. For relatively simple, well defined “consumer subproblems” such as following the recommendations of a GPS navigator or using a shopping bot to find a lower price, I think the answer is clearly yes.

I think the jury is still out when it comes to larger scale and more ambitious problems such as financial planning and life cycle decision making. Here I believe more research is necessary to see how well formal DP solutions such as those provided by the *MaxFi* software actually change individual behavior and improve welfare. In the meantime, I agree with Lusari and Mitchell (2014) who note that we still know relatively little about the degree to which improving financial literacy will actually improve welfare, since “To date, however, relatively little has been learned about whether more financially knowledgeable older adults are also more successful at managing their resources in retirement, though the presence of scams among the elderly suggests that this topic is highly policy-relevant.” (p. 26). Further, I agree with their conclusion that “Relatively little is known about the effects of financial advice and whether it can improve financial decision making. Some preliminary evidence suggests that financial counseling can be effective in reducing debt levels and delinquency rates. In practice, however, most people continue to rely on the help of family and friends for their financial decisions.” (p. 37).

However given the widespread evidence of potentially suboptimal individual decision making, I believe there is a big potential market for algorithms and decision assist tools that can help individuals make better decisions, and I applaud Larry Kotlikoff for his pioneering and ambitious work to undertake practical implementation of DP to improve life cycle decision making. It is a much more sophisticated application of the “nudge” idea of Thaler and Sunstein (2008). But there is an important gap in the literature: we need better tools to scientifically evaluate the effect of this advice, both on objectively measurable outcomes, but also on the most important but difficult to measure variable: individual welfare. In the next section I consider the role of DP for improving decisions by firms, where the key welfare measure is potentially much easier to observe: profits.

4 Applications of DP to improve firm decision making

In this section I discuss several examples where DP has been applied to improve firm decision making, as well as several other academic studies of firms that have not adopted DP but where it seems likely the adoption of DP-based policies could increase discounted profits. To the extent these are credible examples, they cast doubt on the default assumption in economics, namely that firms behave “as if” they maximize expected discounted profits and have unbounded rationality, and suggest that DP has great promise to improve decision making for a wide range of problems.

Unfortunately, similar to the previous section, I am only aware of a few examples where DP is actually used. A skeptical economist might reply that DP is not used precisely because it is unnecessary: firms are able to maximize their profits without the help of DP. Despite Herbert Simon's fundamental contributions to boundedly rational theories of the firm, economists have larely ignored it and remain more comfortable with the default "as if" assumptions of profit maximization and unbounded rationality.

It is reasonable to presume that the stakes for adopting profit-maximizing policies are quite high, so successful, innovative firms that can muster the resources necessary to solve difficult decision problems would appear to be among the early adopters of DP. On the other hand, if DP is such a powerful tool, it seems strange that I had so much difficulty in finding many confirmed examples where it is actually used. Though I have not done a systematic survey, some firms that are likely adopters of DP appear reluctant or unable to speak about it. For example Patrick Bajari, Chief Economist at Amazon reports "I'd love to help out here, but I can't really talk about the IP of our company."

On the other hand, I can confirm that for whatever reason, DP is *not* used by some of the largest, most successful and sophisticated firms who we might expect would be among the most eager early adopters of this technology. For example Benjamin Golub, a PhD in economics from MIT and Chief Risk Officer of Blackrock reports that "Regarding DP, I am not familiar with any uses of it on my side of Blackrock. People use recursive algorithms for various purposes, but I don't believe that is a form of DP. Similarly, there is some use of multi-period optimization, but again I don't think that counts as DP." Blackrock does invest heavily in other forms of high technology including ML for default prediction and other applications. Indeed one of my PhD students, an expert in ML, was hired by Blackrock to join a team developing advanced algorithms to predict mortgage default. However to my knowledge they do not use any form of DP.

Most of the applications of DP that I am aware of are in *engineering* such as network routing Mak, Cheung, Lam, and Luk (2011), power management in cloud computing centers Zhang, Wu, Chen, Cai, and Peng (2017), and several other examples. Powell (2010) discusses several real world applications of DP and notes that "Our experiences using approximate dynamic programming have been driven by problems in transportation with decision vectors with thousands or tens of thousands of dimensions, and state variables with thousands to millions of dimensions. We have solved energy problems with 175,000 time periods. Our applications have spanned applications in air force, finance, and health." Below I discuss a specific application of DP to dynamic assignment of locomotives at a major U.S. railroad.

The limited number of known applications of DP by firms may simply reflect the limitations of DP including the inability to even mathematically formulate some of the most challenging problems that firms confront. Much of the complexity of running a firm stems from the fact that they are complicated multi-person teams, and perhaps some of the difficult challenges involve how to properly incentivize managers and employees to maximize the objective of the firm as a whole. However the design of incentives is one of the area economists have the most expertise in, and indeed my first DP "success story" below is an example where DP helped a firm design a more incentive-efficient compensation plan.

I believe that another problem limiting the application of DP is that for many firms, the objective function that they are maximizing is rather fuzzy and is not entirely clear. The default assumption in economics is that public firms seek to maximize their market value, and their market value equals the expected stream of future dividends, discounted at an appropriate risk adjusted discount rate. But a substantial amount of empirical work in finance on the "excess volatility" of stock market valuations raises substantial doubt as to whether the market value of the firm equals its "fundamental value" i.e. the expected present value of its future dividend stream. To the extent there is a large and systematic "noise" component of stock market valuations, it is no longer clear if a manager's objective function should even be to maximize expected discounted dividends, even if all the other incentive problems involved in operating a firm could be solved.

Further, the theory of the firm ignores potential differences in the behavior of public and private firms. Gupta and Rust (2018) show that if the owner of a private firm is a single individual whose main source of income is the dividend of the firm he/she owns, the appropriate objective is expected discounted utility maximization, not expected discounted profit maximization. The standard *consumption smoothing* motive implied by discounted utility maximization by risk averse consumers implies that owners of private firms should engage in *dividend smoothing* which is inconsistent with the policy of maximizing discounted expected dividends, which generally results in highly volatile dividend policies, including paying zero dividends for long stretches in order to finance investment and accelerate the growth of the firm. Yet paradoxically, there is empirical evidence that public firms tend to smooth dividends *more* than private firms, see e.g. Leary and Michaely (2011). These are among some of the numerous unsolved puzzles about firm behavior that suggests the objective of many firms is not necessarily to maximize the present discounted value of dividends, and that our understanding of firm behavior (including exactly what objective they are trying to maximize) is still quite rudimentary.

Even though we may lack a good global model of how firms behave, my own experience suggests that if a manager believed that a DP-based recommendation system could help increase expected discounted profits for some part of their business, it would be something they would be very interested in adopting provided such a system did not cost too much, would not lead the firm to undertake excessive risks, radically change the firm's operations, or entail a counterintuitive strategy that would be difficult to explain to its shareholders. Thus in this section we focus on applications where DP has been used to make *incremental changes* in firm policymaking, rather than radical policy changes such as replacing experienced managers who make the key judgements on how best to set the general strategy of the firm with a computer algorithm. Just as individuals are ok with using cruise control and other more limited computerized driving assist tools but not yet ready to ride in a fully autonomous driverless vehicle, many firms also seem open to using DP for recommendations on more limited, well defined decisions but would never trust a DP algorithm to run the entire firm.

Thus the examples below are all examples of the application of the principle of decomposition, where DP is used for fairly circumscribed, well-defined subproblems of the overall decision problem that the firm confronts. Further DP is only used as a *decision assistant,* to provide *recommended decisions* that can be overridden whenever they conflict with human judgement and intuition.

A final barrier to the implementation of DP is the lack of tools and methods to validate whether the investment to develop or acquire a DP-based decision tool really generates sufficient incremental revenue or profit to justify adopting it. I provide a few examples of validations that I am aware of, but this remains an area where better scientific methods may help increase the credibility of DP-based decision tools and extend the range of real world applications.

4.1 Application of DP to employee compensation

Misra and Nair (2011) is the most successful *demonstrated* real-world application of DP and structural estimation that I am aware of. It grew out of an academic study of the sales force of a company that produces contact lenses. The authors obtained monthly data on 87 sales agents who are assigned to sell large quantities of contact lenses to eye doctors in different regions of the US. The agents were paid a salary plus a commission that is recalculated each quarter. The salary is paid monthly and the commission each quarter. Commissions are earned on sales exceeding a quota but less than a ceiling. Each quarter the sales "state variable" used to determine commissions is reset to zero, and the firm had a complicated ratcheting formula that adjusted the sales quota and ceiling based on past sales performance.

An initial reduced-form empirical analysis of the data revealed spikes in sales when agents were close to making their quarterly sales quota (after which they would start receiving commmissions), but sales were lower early in each quarter, suggesting the "possibility of shirking early in the quarter" when they were far from reaching their quota. Further for some agents sales would also fall at the end of the quarter "perhaps because the agent realized a very large negative shock to demand early in the quarter and reduced effort, or because he 'made quota' early enough, and hence decided to postpone effort to the next sales-cycle" (p. 219). Finally the authors also provide evidence of more sophisticated forms of agent shirking in response to the firm's ratcheting policy for quotas and commission celings. As they note "Ratcheting can help the firm fine-tune quotas to the agents true productivity or territory potential; but can result in harmful distortions to behavior if forward-looking agents shade current effort in order to induce favorable, future quotas." (p. 214).

This first stage analysis lead the authors to formulate a DP model of the dynamic allocation of sales agent effort over time, as a best response to the incentives of the firm's commmission scheme. The reduced-form analysis suggested distortions in sales effort in response to the peculiarities of the firm's incentive plan. Using a structural analysis and a DP model of the dynamic allocation of effort, the authors were able to make *counterfactual predictions* of sales effort to different types of incentive plans that the firm may not have considered. A brilliant feature of their structural DP analysis is that they were able to estimate structural parameters characterizing *agent-specific preferences* for earnings versus sales effort, even though sales effort is unobserved. They were able to do this by making the modeling assumption that monthly sales (which they do observe for each agent) is a linear function of effort and the coefficient on effort is normalized to 1.

Their structural analysis revealed that their model provides a good fit to the patterns they observed in the sales data that their reduced-form analysis revealed, and provided strong evidence that a poorly designed commission scheme was the cause of the distortions they documented. Using their structural model the authors then solved for an optimal commission scheme, i.e. they used DP to maximize the discounted profits of the contact lens division by optimizing over different parameters indexing different possible compensation plans taking the agent's "best responses" to each possible compensation plan explicitly into account.[22] The authors were able to use DP to solve for *agent-specific compensation plans* since they had estimated separate preference and productivity parameters for each of the firm's 87 sales agents.

However the firm had concerns over possible morale/discrimination problems if it implemented agent-specific compensation plans, and these morale concerns were not explicitly accounted for in the authors' structural model. Instead, the authors proposed a single uniform compensation plan that applied to all of the agents and recommended doing away with with the quotas, ceilings, and periodic ratcheting of the parameters of this plan. Even though this simplified compensation plan is not fully optimal, the firm decided to adopt it as a new compensation plan for its employees starting in January 2009. A before/after comparison of the firm's profits revealed that "Under the new plan, revenues to the firm increased by about $12 million incremental per year (a 9% improvement overall), indicating the success of the field

[22] The use of parameterized compensation plans may result in schemes that are not fully optimal since they are parametric and the authors did not invoke the "revelation principle" of Myerson (1981) to search over the set of all possible incentive-compatible compensation plans.

implementation" (p. 213). Though it is possible to criticize a simple before/after comparison as not adequately controlling for possible macro shocks, note that the new plan earned more even though it was implemented during the depths of the Great Recession. Thus it is likely their estimates of the gains are under-estimated and thus a conservative estimate of the actual improvements in profitability in a steady-state scenario.

4.2 Application of DP to locomotive allocation and fleet sizing

Another highly successful application of DP is reported in Powell et al. (2014) who used RL/ADP to determine the optimal size of the locomotive fleet and provide optimal dynamic scheduling of locomotives to trains, accounting for dynamic schedule changes and locomotive maintenance including shop delays and unexpected locomotive failures for Norfolk Southern railroad. They note

> "To our knowledge, it is the first optimization-based model of locomotives for a North American freight railroad that calibrates accurately against historical data, making it useful as a tool for fleet sizing, one of the most demanding strategic planning problems. The technology allows locomotives and trains to be modeled at an extremely high level of detail; train delays can be modeled down to the minute. The model can simultaneously handle consist breakups and shop routing while also planning the repositioning of locomotives to other terminals. In addition, it can handle uncertainties in transit times, yard delays, and equipment failures in a simple and intuitive way. The methodology is based on a formal mathematical model, which guided the design of rigorous algorithms; as a result, it avoids the need for heuristic rules that have to be retuned as the data change."

Prior to the development of their model, which they call "locomotive assignment and routing system" (LARS) Norfolk Southern relied on a simple linear regression model to estimate the size of their locomotive fleet and noted that "although senior management acknowledged the limitations of the linear regression model for forecasting, it had no better alternatives." The gains from adopting the DP-based LARS model are potentially huge since they note "We cannot underestimate the importance of determining the correct fleet size. Norfolk Southern currently runs a road fleet of over 2,000 locomotives and a new locomotive costs well over $2 million. An insufficient number of locomotives results in deteriorating customer service as train delays increase and leads to a loss of competitiveness in key markets. In contrast, an excessive number of locomotives is a large capital expenditure and is expensive to maintain or store."

In terms of the algorithms, though the authors used RL/ADP due to deal with the curse of dimensionality in allocating 2000 locomotives on a minute by minute basis, they did not "train" the LARS system in real time and their algorithm is not "model-free". Instead LARS is an example of *model-based ADP* where value function iterations are used based on stochastic simulations of a detailed simulation model of system-wide demand for locomotives including times locomotives spend in the shop for maintenance (either planned or unplanned). This latter model, called PLASMA, required a separate multi-year development effort in which the team "carefully calibrated the model against several years of historical performance. This required us to painstakingly examine detailed assignments and compare high-level performance metrics. The process required that we iteratively identify and correct data errors, enhance the model, and make occasional improvements to the basic algorithm."

The benefit of this substantial investment is that "LARS has been used for strategic fleet sizing for several years and has become an integral part of the companys network and resource planning processes." Though the article does not provide an evaluation of how much LARS improved profitability at Norkfolk Southern, the authors note that LARS was implemented just before the Great Recession and the company found it effective in helping downsize the fleet during the recession and then to increase it again in 2010 as the economy started to recover. The authors acknowledge that "These benefits have a cost. LARS requires substantial effort to prepare the data and calibrate the model. However, the impact on operating costs, performance, and network robustness is dramatically large relative to the cost of developing and maintaining the system."

4.3 Potential DP application: replacement of rental equipment

In this and the remaining subsections I discuss a number of academic studies that provide promising examples where DP *could be applied* to help improve firm profitability, but for whatever reason firms have not actually adopted DP so far in their actual operations, unlike the two examples discussed above. In this section I consider the problem of *optimal replacement rental equipment* and discuss two studies: the Cho and Rust (2010) study of rental car replacement and the McClelland and Rust (2018) study of replacement of construction equipment. Unlike the problem of managing an entire fleet of locomotives, the optimal replacement problem takes the fleet size and composition as given and considers the much simpler subproblem of determining the optimal time to replace an old car or piece of construction equipment with a new one. As such, this is another application of the decomposition principle, since changing the timing of replacement need not alter overall fleet size or composition, or even the allocation of different pieces of equipment to different rental locations.

Cho and Rust (2010) obtained panel data on a sample of rental cars from a large Korean rental car company. The data include maintenance costs, new car costs and sales prices received when the company replaced an old rental car with a new one, and a complete day by day history of each car over its lifetime at the firm, including whether it is in the lot waiting to be rented, in the shop for maintenance, or in either a short term or long term rental spell. A reduced-form empirical analysis of these data enabled the authors to develop a stochastic simulation model that can replicate the behavior of the fleet under the company's *status quo* replacement policy, which was to replace its cars after approximately 2.7 years or 75,000 kilometers. The authors found the rental car business is generally quite profitable: the mean internal rate of return the company earned on a typical rental car was approximately 50%.

Yet the authors noted two major puzzles about the firm's behavior: 1) given the rapid depreciation in new car prices, it is very costly for the firm to replace its cars so rapidly, 2) the rental company does not price disciminate by charging a high rental rate for newer cars, but instead adopts a *flat* rental rate schedule where the rental rate does not depend on the age or odometer value of the vehicle. Cho and Rust (2010) used their 3 state semi-Markov econometric model of rental car states and a standard solution algorithm (policy iteration) to solve for the optimal replacement policy for individual cars in the company's fleet under the assumption that if the optimal policy entails keeping rental cars longer than the company currently keeps them, that rental rates of older cars would be discounted to induce customers to rent them. The optimal replacement policy from the solution to the DP implied keeping rental cars roughly twice as long as the company does under its existing replacement policy. Doing this substantially increases discounted profits, by 18% to 140% depending on the specific make/model analyzed.

The rental car company was intrigued by these findings but was fearful that a policy that discounts older cars would cause consumers to substitute from new car rentals to rental of older cars, potentially lowering revenues earned from rental of newer vehicles and damaging the reputation of the company. Total rental revenues will mechanically fall at a given location if the firm keeps rental cars longer and discounts the rates of older cars to incentivize customers to them (assuming a fixed fleet size at the location), but the econometric model predicted that replacement costs would fall even more, so total profits increase. Cho and Rust (2010) succeeded in convincing the company that if the appropriate discounts were chosen, total revenues could actually *increase* due to price discrimination which would be a double win for the firm.

These intuitive arguments, combined with the realistic simulation results, convinced management to test the predictions using a controlled field experiment where the company chose 4 rural "treatment locations" where the timing of replacements were delayed but rental rates on cars older than 2 year old were discounted by 20%. The profitability of these locations were then compared with the profitability at 6 "control locations" where the company maintained its *status quo* replacement and rental rate policy. The results of the experiment revealed that there was minimal substitution from new cars to older cars due to the 20% price reduction, but the total utilization (and thus rental revenue) from older rental cars increased substantially: average rental revenue at the 4 treatment locations was 12% higher than at the 6 control locations. Thus, with the increased rental revenue and substantial decrease in replacement costs, the field experiment convincingly validated the predictions of the DP model and confirmed that the rental company's existing replacement policy is suboptimal.

McClelland and Rust (2018) analyzed a data set of several thousand construction machines (excavators, high reach forklifts, scissor lifts, skid steers and telescopic booms) that are rented to construction companies and other firms in much the same way that rental cars are rented to consumers. They confirmed that the "flat rental puzzle" for rental cars is also present for rental machines: the rental companies do not discount the rental rates on older machines to incentivize its customers to rent them. As a result, their analysis finds that effective rental utilization (i.e. the fraction of the time a machine is rented vs waiting in the lot to be rented) declines with the age of the machine, similar to what Cho and Rust (2010) found for rental cars. However unlike rental cars, McClelland and Rust (2018) found big variations in replacement policy for different types of machines: some were sold 6 years after their initial acquisition, but others were kept for 8 or more years. Using expected monthly maintenance costs and rental revenues, McClelland and Rust (2018) uncovered obvious evidence of suboptimality in replacement behavior even before calculating the DP solution. For example for scissor lifts, they find that average monthly profit turns negative after the machines are older than 6 years, but the mean age at which these machines are sold is 7 years. Thus, unlike rental cars, they find that some types of rental equipment are kept *too long.*

However by using DP they were able to uncover a more subtle form of suboptimality in the replacement behavior of the company they studied. The prices of new construction equipment moves in a highly *pro-cyclical* fashion, following the pro-cyclical pattern in overall construction spending: firms try to shed construction equipment in a recession when they except a period of low construction investment, and they rush like lemmings to buy new equipment at the start of a boom period when they expect a period of high sustained construction spending. The prices of used machines, however, tends to move on a counter-cyclical fashion, tending to be high relative to new machine prices (OEC) in recessions when relatively more firms choose to keep rather than sell their older machines, but lower in booms when firms tend to sell their older equipment to buy new replacements. These predictable swings in the relative costs of new and used machines imply a pro-cyclical variation in replacement costs for some types of machines: with replacement costs being relatively high in economic booms and low in recessions. The DP algorithm is able to identify profit opportunities by effectively arbitraging these predictable swings in replacement costs, resulting in a counter-cyclical optimal replacement policy. Overall equipment replacement for the US economy as a whole is overwhelmingly pro-cyclical. Thus, under the assumption of no financial "liquidity constraints" (which is valid for most of the large equipment rental companies) the DP solution of McClelland and Rust (2018) identified profit opportunities from the adoption of a "contrarian" counter-cyclical optimal replacement investment strategy. The potential increase in expected discounted profits vary by type of machine and region considered, but ranged from 10% to 25%.

4.4 Potential DP application: decision making in professional sports

The popular book *Moneyball* by Lewis (2003) described how the use of *sabremetrics* i.e. the analysis of baseball statistics and the use of simple decision models to improve decision making could have significant payoffs, in particular helping baseball owners to assemble winning teams at far lower cost. According to *Wikipedia* "The central premise of Moneyball is that the collective wisdom of baseball insiders (including players, managers, coaches, scouts, and the front office) over the past century is subjective and often flawed. By re-evaluating the strategies that produce wins on the field, the 2002 Athletics, with approximately US $44 million in salary, were competitive with larger market teams such as the New York Yankees, who spent over US $125 million in payroll that same season. Because of the team's smaller revenues, Oakland is forced to find players undervalued by the market, and their system for finding value in undervalued players has proven itself thus far. This approach brought the A's to the playoffs in 2002 and 2003."

Moneyball and the sabremetric approach had tremendous impact and was widely imitated by other major league baseball teams who hired their own full time sabremetric analysts. The success of sabremetrics provides evidence that baseball owners could not have been behaving "as if" they were unboundedly rational profit maximizers. However sabremetrics is not the same as DP: could owners do even better by using DP to find better strategies for playing game and assembling winning teams?

Though DP has proved extremely effective in generating superhuman strategies in two player board games such as chess or Go, the complexity of multiplayer physical games such as baseball, basketball or football (i.e. the large number of variables required to capture the current state of the game) combined with the curse of dimensionality makes it unlikely that DP can be used to provide better strategies for playing individual games, or the vastly more complex "meta game" of how to recruit/draft players to assemble a winning team subject to a budget constraint. However, appealing to the principle of decomposition, there may be simpler "subgames" for which DP can be applied to solve and might be useful for improving decision making.

For example, Romer (2006) analyzed "a single narrow decision — the choice on fourth down in the National Football League between kicking and trying for a first down … Play-by-play data and dynamic programming are used to estimate the average payoffs to kicking and trying for a first down under different circumstances. Examination of actual decisions shows systematic, clear-cut, and overwhelmingly statistically significant departures from the decisions that would maximize teams chances of winning." (p. 340). The reason that DP is required is because the future consequences of missing a field goal versus trying either for a touch down or first down are quite different and depend on the current "state" (yard line where the ball is currently at). Failing to get a touchdown or first down leaves the ball at whatever yard line the runner with the ball is tackled at. But failing to get a field goal requires the team to do a kickoff on the next round, which is better for the opposing team if the ball is sufficiently close to the 0 yard line.

Romer uses DP to calculate optimal 4th down strategies on the assumption that the football team's objective is to maximize the probability of winning the game. He finds that football teams are much more likely to try to kick when the DP decision rule prescribes going for it, and thus, "Teams' actual choices are dramatically more conservative than those recommended by the dynamic-programming analysis." (p. 354). Romer speculates that one possible explanation for this is that the objective function for a football team is not simply to maximize the proability of winning the game: "the natural possibility is that the actors care not just about winning and losing, but about the probability of winning during the game, and that they are risk-averse over this probability. That is, they may value decreases in the chances of winning from failed gambles and increases from successful gambles asymmetrically. If such risk aversion comes from fans (and if it affects their demand), teams choices would be departures from win maximization but not from profit maximization." (p. 361).

Though the optimal behavior predicted by DP clearly depends on what the actual objective function is, and the initial reaction to his findings was met with a bit of skepticism from NFL players and coaches there is evidence that football teams have noticed Romer's analysis and have changed their behavior: they are more likely to go for it than to kick. The late great football coach Bill Walsh said "I think (coaches) can tend to be too conservative. They can tend not to trust themselves — in a sense, take the easy way out and, in this case, punt the ball." Indeed in Superbowl 52, the Philadephia Eagles decided to go for it on the 1 yard line, and won the game against the New England Patriots. A *New York Times* article on February 2, 2018 discussed how the Eagles' coach Doug Pederson "Followed the Numbers to the SuperBowl" and noted that "The Eagles, by going for it, improved their probability of winning by 0.5 percent. Defending his decision (again) at a news conference the next day, Pederson cited that exact statistic."

4.5 Potential DP application: inventory management and commodity pricing (steel)

Hall and Rust (2018) analyzed high frequency inventory and pricing data from a steel service center (SSC). These are essentially large warehouses that hold different types of steel products such as various sizes of coil steel and plate steel. A SSC does minimal production processing and mainly functions as an inventory storage depot to provide more immediate access to steel by final customers. SSCs make their profits in a very simple manner: by purchasing large quantities of different steel products at low prices and selling smaller quantities to their retail customers at a markup. Thus, there are two key aspects to the profitability of a SSC: 1) success in the timing of its steel purchases (i.e. its ability to time the market in order to "buy low and sell high"), and 2) its ability to quote the highest possible price to different customers. Unlike other commodities, there is no centralized market for steel such as the New York Mercantile Market or the Chicago Board of Trade where other commodities such as wheat or pork bellies are traded. As Hall and Rust (2003) discuss, the steel market can be described as a highly fragmented and decentralized "telephone market" where customers must engage in costly search by calling individual SSCs to ask for price quotes over the phone. Surprisingly, even in 2018 very few steel products have their prices publicly displayed on the web, a puzzle that the model of Hall and Rust (2003) is designed to address. Because prices are individually negotiated over the phone, two different customers buying the same type of steel on the same day can pay different prices: firms engage in (legal) third degree price discrimination in the steel market.

Hall and Rust (2007) formulated a DP model of the optimal inventory purchasing/holding and retail pricing problem that maximizes the expected discounted profits of an SSC. They showed that the optimal strategy takes the form of a generalized (S,s) rule, where $S(p,x)$ is a function that represents the optimal or desired level of inventory that the firm should hold, and $s(p,x)$ is an *order threshold.* The $S(p,x)$ and $s(p,x)$ functions depend on the wholesale price of steel p and a vector x of other variables that are useful for predicting future prices and demand for steel. Hall and Rust (2007) show that when the stochastic process for wholesale prices $\{p_t\}$ is a stationary $AR(1)$ process (and hence mean reverting), both of the S and s thresholds are declining functions of p enabling the implied optimal speculative purchasing strategy for steel to capture the intuitive property of stocking up when prices are low, but not replenishing inventories when wholesale prices are high. As noted in the introduction, Hall and Rust (2007) appealed to the principle of decomposition by treating each of the 9000 products the SSC sells as separate "subfirms" that solve a DP problem where (q,p,x) are the relevant state variables instead of a single joint DP problem for all 9000 products simultaneously. This decomposition is valid provided a) there are no storage constraints for the plant as a whole, and b) there are no financial constraints that affect the timing or amount of steel the firm can purchase. If either of these assumptions are violated the inventory holding and purchase decisions of different steel products may be interdependent and it may be necessary to solve the much harder joint problem. But for the particular firm that Hall and Rust (2018) analyzed conditions a) and b) appear to hold and the decomposition of the firm's overall profit maximization into separate subproblems appears to be a good approximation to reality.

When there is a fixed cost $K > 0$ to placing an order to buy more steel, it is not optimal to always maintain the quantity of steel q equal to the optimal target value $S(p,x)$. Instead if current inventory q is above the order threshold $s(p,x)$ then no new order is placed, but if $q < s(p,x)$ it is optimal to order an amount $q^0 = S(p,x) - q$ to return current inventory holdings back to the optimal target value $S(p,x)$.

The model also predicts the optimal price setting strategy by an SSC. Pricing can be viewed as a take it or leave it offer by the firm that reflects information the company learns about a customer such as the amount of steel demanded and where it is to be shipped. If customers are engaging in costly search, their purchase strategy is governed by a reservation price strategy as Hall and Rust (2003) demonstrate. Thus, the customer will buy if the SSC quotes a take-it-or-leave it price below their reservation price. Though the SSC will generally not know a customer's reservation price, if it knows the distribution of reservation values, a formula for the optimal customer-specific take-it-or-leave-it price quote can be calculated as a markup over the replacement cost of the desired quantity of steel the customer wants to buy using the same logic as Myerson (1981) for setting the optimal reservation price in an auction with a single bidder.

The empirical analysis by Hall and Rust (2018) estimates the parameters of the stochastic process for wholesale prices and the distribution of consumer reservation values via structural estimation by the method of simulated moments (MSM) which is able to deal with the *endogenous sampling problem* i.e. the SSC only records the prices of steel purchases on the days it buys steel, not on days it doesn't buy. Despite this sampling problem, the structural MSM estimator is able to uncover the parameters of the AR(1) process for wholesale prices and other relevant structural parameters. Unfortunately, the structural model is strongly rejected via standard specification tests: the firm's actual purchasing and pricing decisions are not well approximated by the (S,s) rule. Using stochastic simulations that condition on prices the firm was actually able to purchase at, the authors compare the actual profits earned over the period 1997 to 2004 for several different steel products with the profits that the firm could have been expected to earn had it followed an optimal (S,s) ordering strategy and adopted an optimal pricing strategy. Even if the firm used a naive, suboptimal fixed markup over the current wholesale price, the authors demonstrate that the (S,s) strategy could increase the firm's profitability by between 12% and 29% depending on the product. These findings suggest that the reason why the (S,s) model is rejected is that the steel company is using a suboptimal strategy for purchasing steel: the (S,s) rule is more effective in recognizing the "turning points" when the wholesale price of steel is lowest and buying larger quantities at those times. As a result, the (S,s) makes significantly more speculative trading profits — i.e. it does a better job of "buying low and selling high" than the steel company does.

Profits can be increased further by adopting an optimal retail pricing strategy using the optimal Myerson markup rule. In separate work Hall and Rust provide evidence that the company's sales agents are failing to extract as much surplus as possible from their ability to engage in 3rd degree price discrimination, though some sales agents appear to be more effective price setters than others. Additional pricing mistakes result from the failure of the firm's sales agents to correctly calculate the replacement cost of the steel they sell. The firm's inventory management system only provides sale agents with the *historical cost* of the steel they sell, but an optimal pricing rule requires setting a markup over the *actual replacement cost of the steel.* This can be calculated using the DP model. Note that the economically relevant measure of the replacement cost is *not* the current wholesale price of steel. Instead it is a 'shadow price' that equals the derivative of the value function with respect to q. This shadow price can fall below the current wholesale price if the firm is overstocked with steel (i.e. if $q > S(p,x)$), and it can be substantially higher than p if the firm is near the re-order threshold, i.e. when $q < s(p,x)$. In addition, information such as steel tariffs can dramatically affect the relevant replacement cost. The steel firm acknowledges that it probably loses significant profits since it is unable to provide its sales agents with information on the relevant replacement costs for the 9000 steel products it sells so they are able to set their retail price quotes correctly.

4.6 Potential DP application: perishable inventory pricing and revenue management

My final example of the potential benefits from DP comes from the area of revenue management and dynamic pricing, particularly in the travel industry such as airline and hotel pricing. Unlike steel, the "inventory" held by an airline or a hotel is perishable: if a given steel coil is not sold today, it is still there to be sold to some other customer tomorrow, whereas if a plane takes off with empty seats or a hotel has unoccupied rooms, this unsold inventory can never be sold again. In addition the available capacity of an airline or hotel is fixed in the short run. This implies that airlines and hotels face strict quantity constraints that a steel company does not face since a steel company can accept an order even if it is stocked out by placing a quick order to restock and delivering the item to the customer after a delay. As a result, the nature of the optimization problem hotels and airlines face is rather different than the problem the steel company faces, where adjusting inventory in response to demand shocks is much easier and less costly.

Revenue management (also known as "yield management" in the airline industry) focuses on strategies to maximize revenue by allocating available seats (or rooms) at different prices via price segmentation strategies (e.g. charging different prices for business vs leisure travelers) based on differential willingness to pay. Some revenue stategies are quantity-based (i.e. "protecting" certain seats or rooms from early booking so they are available for last minute booking by customers with high willingness to pay) and others are price-based *dynamic pricing* strategies that adjust prices and do not directly control quantities.

Phillips (2005) provides a comprehensive introduction to the literature on revenue management. He notes that despite the fact that pricing decisions "are usually critical determinants of profitability" they are "often badly managed (or even unmanaged)." (p. 38). Computerized revenue management systems (RMS) originated in the 1980s following airline deregulation when American Airlines was threatened by the entry of the low-cost carrier PeopleExpress.

> "In response, American developed a management program based on differentiating prices between leisure and business travelers. A key element of this program was a 'yield management' system that used optimization algorithms to determine the right number of seats to protect for later-booking full-fare passengers on each flight while still accepting early-booking low-fare passengers. This approach was a resounding success for American, resulting in the ultimate demise of PeopleExpress." (p. 78).

However surprisingly, there is *no mention of DP* in Phillips' book. Though the book conveys a great deal of insight and understanding of revenue management, and contains many intuitively appealing heuristic principles such as *Littlewood's Rule* (a pricing rule that arose from academic work on static revenue management models in the 1970s), it fails to provide tools for handling the stochastic dynamic nature of booking dynamics. A further challenge is that pricing decisions must be done in real time. A small hotel may need to adjust 1000 prices per day and a small airline, 500,000 prices daily. The high dimensionality of these pricing decisions comes from the fact that hotels and airlines are constantly adjusting *future prices* for departures or occupancies on various flights and different classes of rooms as much as a year into the future. In the past prices may have been adjusted only once per day, but now price adjustments are being made more and more frequently. As Phillips notes "The Internet increases the velocity of pricing decisions. Many companies that changed list prices once a quarter or less now find they face the daily challenge of determining which prices to display on their website or to transmit to e-commerce intermediaries. Many companies are beginning to struggle with this increased price velocity now — and things will only get worse." (p. 55).

Though there is a rather large academic literature that uses DP to solve the revenue management problem including the pioneering work of Gallego and van Ryzin (1994), and though the size of the revenue management industry has mushroomed to over $10 billion per year, I am not aware of specific commercial evenue management firms that are actually using DP to calculate optimal price or quantity allocation recommendations to their customers. As McAfee and te Veld (2008) note "At this point, the mechanism determining airline prices is mysterious and merits continuing investigation because airlines engage in the the most computationally intensive pricing of any industry. The techniques used by airlines are likely to be copied elsewhere as the ability of companies to condition price offers on other considerations grows with larger databases and enhanced processing power." (p. 437).

Regardless of whether DP is specifically used to determine revenue management strategies, the industry appears to have been highly successful. Gallego and van Ryzin (1994) note that "The benefits of yield management are often staggering; American Airlines reports a five-percent increase in revenue worth approximately $1.4 billion dollars over a three-year period, attributable to effective yield management." Anderson and Kimes (2011) claim that "In many instances RM used in the hotel industry has been shown to increase revenue by 2 to 5 percent." The *Wikipedia* page on revenue management describes highly sophisticated analyses of consumer demand are improving the ability of these systems to extract consumers' surplus

> "Realizing that controlling inventory was no longer sufficient, InterContinental Hotels Group (IHG) launched an initiative to better understand the price sensitivity of customer demand. IHG determined that calculating price elasticity at very granular levels to a high degree of accuracy still was not enough. Rate transparency had elevated the importance of incorporating market positioning against substitutable alternatives. IHG recognized that when a competitor changes its rate, the consumer's perception of IHG's rate also changes. Working with third party competitive data, the IHG team was able to analyze historical price, volume and share data to accurately measure price elasticity in every local market for multiple lengths of stay. These elements were incorporated into a system that also measured differences in customer elasticity based upon how far in advance the booking is being made relative to the arrival date."

Given this high degree of sophistication, what additional value added might be provided from DP? Two recent academic studies, Williams (2018) and Cho, Lee, Rust, and Yu (2018) use DP models to evaluate the pricing decisions of a specific airline and hotel, respectively. In effect, these papers developed their own DP-based revenue management systems and compared how well the DP models are able to predict the actual prices set by the airline and hotel, respectively. Since the airline and hotel rely on recommended prices from professional RMSs, these studies provide external evidence on the question of how closely the price recommendations of commercial RMSs are to optimal prices calculated by DP.

Williams studies prices set by a particular airline on several of its monopoly routes and shows his structural DP model of dynamic pricing closely matches the actual prices set by this airline. While this does not prove that the RMS actually uses DP to calculate its recommended prices (the RMS may use heuristic principles that nevertheless enable it to closely aproximate a DP solution, corresponding to the "as if" argument for optimality), Williams' analysis provides considerable insight into the revenue and efficiency gains from the use of dynamic pricing. The optimal dynamic pricing rule depends critically on the level of unsold capacity and the number of days remaining to take off as the key state variables. Williams finds that price paths vary dramatically depending on how quickly capacity fills up: "If a sell out becomes more likely early on, the firm greatly increases prices which saves seats for business consumers who will likely arrive later." (p. 28). On the other hand, when there is a sufficiently high chance of unsold seats on the departure date, the optimal strategy can entail significant price cuts. Williams concludes that "By having fares respond to demand shocks, airlines are able to secure seats for late-arriving consumers. These consumers are then charged high prices. While airlines utilize sophisticated pricing systems that result in significant price discrimination, these systems also more efficiently ration seats." (p. 47).

Cho et al. (2018) studied the pricing decisions of a hotel that uses the IDeaS RMS, a subsidiary of SAS. The hotel's revenue manager reports that she usually follows the recommended prices from IDeaS, though she is able to override these prices whenever she wants. Though the recommended prices are usually intuitively reasonable, occasionally the system appears to produce unreasonable recommended prices (such as unusually high prices for bookings in August, which is a hot, slow month for this hotel) and the revenue manager overrides them and sets her own value. The IDeaS RMS does not appear to be adaptive in real time to these human initiated price updates: the system tends to keep making the same mistakes until the revenue manager calls the company and asks them to fine tune the pricing algorithm. That said, Cho et al. (2018) find that the predicted prices from their DP model closely match the prices set by the hotel. Unfortunately the data they analyzed did not record situations where recommended prices from IDeaS were overridden by the human revenue manager, so they were unable to determine whether IDeaS or the human revenue manager's prices more closely conformed to the predicted prices from the DP model.

Unlike the analysis of Williams (2018) the hotel pricing study had access to prices charged by the direct competitors of the hotel they studied (hotel 0). The competing prices turned out to be key components of the state variables in the DP model of Cho et al. (2018). Empirical analysis of the hotel prices in this market show that the prices of all the hotels display strong co-movement, a phenomenon the authors refer to as *price following behavior.* While the co-movement could superficially suggest possible price collusion (such as if all the competing hotels are using the IDeaS RMS and the system was helping to coordinate the prices of these hotels), Cho et al. (2018) conclude that

> "hotel 0's price setting behavior is competitive and is well approximated as a best response to the dynamic pricing of its competitors. The strong co-movement in prices in this market is entirely consistent Bertrand price competition among all of the hotels, which are subject to aggregate demand shocks that cause their occupancy rates and prices to move together. We showed that when a hotel expects to sell out, it is optimal for it to increase its price to ration scarce capacity. This can lead a hotel to increase its price far above the price of its competitors, even though under more typical scenarios where the hotel is far from selling out, it is optimal for the hotel to price below its competitors in a manner that closely approximates the price following behavior that we showed provides a very good approximation to the way hotel 0 actually sets its prices."

Note that Cho et al. (2018) empirical analysis *assumes* optimal pricing on the part of hotel 0. The "as if" assumption of optimality is an important identifying assumption for their econometric analysis, since they show that the strong co-movement in hotel prices is the result of common demand shocks, not collusion. However these demand shocks result in a severe price endogeneity problem that makes it very difficult to identify customer demand: on high demand days the hotel are likely to sell out and they raise their prices to ration the demand in the face of fixed capacity, whereas on low demand days the hotels cut prices to try to sell off their unsold capacity. In the absence of a good instrumental variable, OLS estimation of the demand results in an *upward sloping demand curve.* Cho et al. (2018) explain that there are no good instrumental variables to solve this endogeneity problem. However their MSM estimator, combined with the assumption that hotel 0 prices optimally (i.e. via the solution of the DP problem) provides strong identifying restrictions that results in a well behaved, downward sloping expected demand curve for hotel rooms.[23]

As I noted in section 2.1 further work has shown it is possible to relax the maintained assumption of optimality and still obtain consistent estimates of the stochastic demand/arrival process for hotel bookings using a *semi-parametric two step estimator* that relies on non-parametric estimation of the pricing rules used by hotel 0 and its competitors in the first step, followed by a MSM estimator that finds values of the parameters for the stochastic demand/arrival process for hotel bookings in the second step that best match a set of observed moments for pricing and bookings for these hotels. Then given the estimated demand parameters, it is possible to formulate and solve the hotel's DP problem to calculate optimal dynamic price strategies. By relaxing the optimality assumption, this two step estimation approach enables us to *test for optimality* rather than to assume it. Initial results by Cho et al. (2018) indicate that when the assumption of optimality is relaxed, the DP-based pricing strategy is significantly hotel 0's existing pricing strategy. Simulations comparing the two strategies confirm that the pricing strategy from the DP model results in significantly higher profits. In principle these predictions may be testable, either via before/after field tests, or controlled field tests similar to the ones reported in Cho and Rust (2010) discussed in section 4.3.

5 Conclusion

DP is an extremely powerful tool for solving a huge class of sequential decision problems under uncertainty. In fact it is hard to think of problems that can't be formulated and solved using DP. The flexibility and power of DP has revolutionized the way we do economics, and it has provided a key principle that has produced some of the most impressive achievements in the field of artificial intelligence. Despite this, I have only been able to come up with a few examples where DP is used and has resulted in demonstrable improvements in *real world* decision making. What is the explanation for this paradox?

Of the three possible explanations I listed in the introduction, I think the first explanation, namely, that individuals and firms have unbounded rationality and already behave "as if" they had solved a DP problem, can be immediately dismissed. As Herbert Simon noted in his 1978 Nobel Prize lecture sufficient evidence against the assumption of unbounded rationality had already been accumulated since the 1950s. Since

[23] Note that there is stochastic arrival of customers who book rooms in advance of occupancy. So the notion of a fixed demand curve for hotels is too simplistic. Instead, demand should be viewed as a *price-dependent stochastic process.* The MSM estimator developed by Cho et al. (2018) can uncover the parameters of this stochastic process which includes parameters for an underlying discrete choice model of which hotel a give customer will book at given they want to book at one of the competing hotels, taking into account the attributes and prevailing prices of all of the hotels at the time they book.

then, we have even more evidence of suboptimal decision making by individuals and firms, and additional examples have been provided in this article. Yet we cannot immediately conclude that this implies that individuals and firms behave stupidly, or that they will necessarily benefit from the advice of computerized algorithms solving DP problems. The following characterization of the nature of suboptimality of human decision making by Hutchinson and Meyer (1994) provides the right perspective

> "when compared to normative sequential choice models, humans are likely to perform less well because the processes of forward planning, learning, and evaluation have a number of inherent biases. From a broader perspective, however, one can argue that optimal solutions are known for a relatively small number of similar, well-specified problems whereas humans evolved to survive in a world filled with a large and diverse set of ill-specified problems. Our 'suboptimality' may be a small price to pay for the flexibility and adaptiveness of our intuitive decision processes." (p. 379).

I believe the key reason for the limited number of real-world applications of DP is the difficulty in using it to formulate and solve the highly complex, fuzzy, and poorly defined decision problems that individuals and firms typically face on a daily basis, an explanation that Herbert Simon emphasized in his work on AI and bounded rationality over four decades ago. DP has proved very successful as an *academic modeling tool* but it is far more challenging to take things from pencil and paper and actually formulate and solve a host of real-world decision problems as mathematical DP problems.

Given all the empirical evidence it is hard to understand why economists are so wedded to the hypothesis of unbounded rationality. Though DP encapsulates an idealized model of rational behavior that might be viewed as a reasonable starting point for empirical inquiry, economists' love for highly abstract, technical mathematical theories may be the reason for the continued popularity of DP and game theory in economics. Prior to the advent of structural estimation, most economists avoided actually trying to numerically solve DP models, and instead were content with characterizing general properties of optimal decision rules, such as the reservation price property for search problems, or the (S,s) property in inventory problems, or simply proving the existence of a Nash equilibrium of a dynamic game.

The literature on structural estimation of DP models, which began in the 1980s, lead economists to become more invested in numerical solution of DPs because many of the "full solution" estimation methods such as the "nested fixed point algorithm" of Rust (1987) required repeated solutions of the DP while searching for the structure of the underlying decision problem (i.e. preferences and beliefs) for which the optimal decisions from the DP model best fit actual decisions. The structural estimation literature continued to maintain the standard economic assumption of unbounded rationality, and so assumed all agents behave "as if" they can solve arbitrarily complex DP problems. General results on non-parametric non-identification of structural models implied that Bellman's Principle of Optimality *per se* imposed no testable restrictions on behavior unless we are willing to make fairly strong *a priori* assumptions on the structure of agents' decision problems. Thus, not much could be learned about whether real world agents are really capable of solving complex DP problems outside of laboratory experiments where the structure of the decision problem could be restricted and manipulated. Thus the most decisive evidence against the "as if" unbounded rationality assumption comes from the experimental literature, though many economists dismiss these findings on the grounds that laboratory experiments are unnatural settings where subjects are not sufficiently incentivized to behave optimally.

Once economists faced up to the problem of actually solving specific DP problems numerically, they quickly confronted the curse of dimensionality — the exponential increase in computer power required to solve bigger, more realistic DP problems. Compared to the dauntingly complex decision problems individuals and firms confront every day in the real world, the types of DP problems that have been formulated and solved in the literature on structural estimation have been mere "toy problems" that are in many cases far too simplistic to be taken seriously. A substantial body of work in the computer science on computational complexity subsequently proved that the curse of dimensionality was not just an annoyance that would ultimately be overcome by a sufficiently clever algorithm, but rather is an insuperable obstacle to the numerical solution of DP problems that cannot be overcome by *any* algorithm.

Meanwhile DP was discovered by researchers in AI, who realized it provided a very general framework for modeling intelligent adaptive behavior and learning. A substantial literature on machine and reinforcement learning developed in the 1990s can be viewed as providing a new class of stochastic iterative algorithms for solving the Bellman equation for stationary infinite horizon MDPs. Some of these algorithms such as Q-learning also have the property of being *model-free* i.e. they can asymptotically converge to the optimal DP decision rule without prior knowledge of the underlying structure of the decision problem, so long as the algorithms is able to "experience" current period payoffs during the process of making decisions and learning. Thus, some reinforcement learning algoithms are capable of learning intelligent decision making (i.e. optimal DP decision rules) on the basis of experience *in real time.* Q-learning and related algorithms were extended to work with neural networks and other parameteric families of functions for DP problems with many possible states since the iterative learning algorithms could be reformulated as iterative stochastic algorithms for learning the much smaller number of parameters of a neural network which provided a convenient mechansim for extrapolating and interpolating values for states that are rarely encountered in the course of training. This strategy has lead to amazing recent successes such as Google/DeepMind's ability to train a deep Q network to achieve superhuman levels of ability in a range of board games such as chess or Go after just a few hours of training via self-play.

These successes lead to considerable hope that these new algorithms from the AI/reinforcement learning literature (also know as neural DP or approximate DP algorithms) could break the curse of dimensionality and be successful in solving other complex, high dimensional DP problems. However the complexity bounds from computer science show that the curse of dimensionality cannot be broken by *any* algorithm, including the new stochastic algorithms from the reinforcement learning literature. Though the neuro DP algorithms avoid one form of the curse of dimensionality by approximating the value function using a neural network with a relatively small number of parameters, it does so at the expense of another (the curse of dimensionality in finding values of these parameters that globally minimize a nonlinear least squares objective function). The only real way to break the curse of dimensionality is to exploit various types of *special structure* present in some types of DP problems, such as DP problems where the number of possible choices is finite, Rust (1997). I also emphasized the use of the principle of *decomposition* which involves recognizing that in some situations the solution to a large, complex overall DP problem can be decomposed into a solution of a number of smaller more tractable DP problems. The steel inventory problem discussed in section 4.5 is one such example.

Herbert Simon already anticipated all of these conclusions in his 1978 Nobel Prize lecture "Rational Decision-Making in Business Organizations." Simon understood the problem of the curse of dimensionality even though he did not refer to it usng Bellman's colorful terminology. He referred to his own work with Holt, Modigliani, Muth, and Simon (1960) on modeling inventories held by firms and noted "In the face of uncertain and fluctuating production demands, a company can smooth and stabilize its production and employment levels at the cost of holding buffer inventories. What kind of decision rule will secure a reasonable balance of costs? Formally, we are faced with a dynamic programming problem, which generally pose formidable and often intolerable computational burdens for their solution." (p. 351). Simon introduced the term *satisficing* to summarize how individuals and firms that are boundedly rational actually behave when they confront computationally intractable decision problems. He observed there were two main ways to satisfice: a) either via an informal process of trial and error that leads an individual or firm to settle on a heuristic or rule of thumb that appears good enough, or b) by finding an optimal solution to simplified version of the actual problem.

The latter is what academics typically do when they build formal mathematical models, and it is the area where DP has perhaps the most promise for helping individuals and firms make better decisions. That is, instead of trying to tackle the overall decision problems faced by individuals and firms (problems that are probably too difficult to formalize mathematically, much less be able to solve numerically), faster progress can be made by starting less ambitiously, using the principle of decomposition to look for simpler subproblems that are tractable and can be solved using various versions of DP. Indeed, most of the examples that I have discussed in this article only apply DP in this limited sense: to assist individuals and firms in solving fairly narrow and well-defined subproblems of their overall decision problems that are easier to formulate mathematically and solve numerically.

Simon already recognized that computer power and algorithms would improve steadily over time and thus steadily expand the range of real-world problems that could be solved by DP. Yet it is 2018 and the range of known real world applications of DP seems disappointingly small given the immense computer power and decades of research that have produced a myriad of alternative solution methods for DP problems. I believe the biggest constraint on progress is not limited computer power, but instead the difficulty of learning the underlying structure of the decision problem. Individuals and firms are likely to have difficulty understanding their own decision problems and may be unable to communicate to an outside advisor exactly what objective they are trying to maximize. Perhaps the most painstaking task for an academic or a commercial service trying to play the role of "critic" and recommend better decisions the "actor" (i.e. the individual or firm making the actual decisions) is to *understand the structure of the actor's decision problem.* Calculating an optimal solution to the wrong objective, or misspecifying the constraints and opportunities the actor actually confronts may not result in helpful advice to the actor. It is sort of like providing the right answer to the wrong question.

However as a practical matter, approximations occur at every level in the numerical solution of a DP problem. The objective function itself will generally only be approximately correct as well. However when the approximations are sufficiently good, DP-based solutions can serve as the role of critic that can helps the actor make better decisions. I have provided several convincing examples where this is the case. But underlying each of the success stories is a painstaking amount of empirical analysis and modeling to try to find the best possible approximation to the actual decision problem the actor confronts. The highly publicized successes of the Google/Deep Mind "model free" self-training approach to solving complex DP problems are impressive and seductive, but it leads to the temptation that the analyst can simply download their *TensorFlow* software, turn on your computer, turn off your brain, and let their algorithm learn and discover an amazing optimal DP strategy after a few hours of self-training. This approach may work for a limited class of dynamic games where the structure of the decision problem is sufficiently simple and well understood *a priori* and the main objective of learning opponent strategies. It may work well for improving our understanding the double auction game, for example, the equilibria to which has eluded a theoretical or numerical solution for decades and perhaps the most insight has come from human laboratory experiments and computer tournaments such as Miller et al. (1993).

But for most real-world decision problems, the model-free approaches are unlikely to useful. First, it takes too long to train them in a real time situation and their initial performance is likely to be so poor that no actual decision maker would be willing to use them. The only way that DP strategies trained by reinforcement learning methods will prove useful is if they can be adequately trained *offline* before they are first used in real time. However to train them offline, a realistic simulation model of the actual decision environment is required. But developing such a model requires *human effort, creativity and learning.* I am not aware of any machine learning algorithm that can automatically build a model that provides an adequate approximation to reality without any human supervision. Model building is an essentially amazing innate human capability, something our own neural nets are able to do unconsciously and is essential to our internal perception of reality and the basis of human intelligence Eagleman (2011).

Thus, I do not share tStephen Hawking's fear that AI has grown sufficiently powerful to constitute an iminent threat to the future of human race. Perhaps in decades or centuries after algorithms evolve to equal or surpass the level of human brilliance in contructing, revising, and improving mental models of reality. Humans are learning to replicate the type of subconscious model building that goes on inside their brains and bring it to the conscious, formal level – but they are doing this modeling *themselves* since it is not clear how to teach computers how to model. The development of digital computers, statistics, mathematics and numerical methods and DP are all contributing to a confluence that is starting to bring more intelligent types of algorithms, but I do not think we are yet at the threshold of true artificial intelligence.

In the meantime, there are tremendous opportunities for researchers in economics, operations research, engineering, computer science, neuroscience, and artificial intelligence to work together to produce increasely intelligent DP-based algorithms that will have significant practical value to individuals and firms by helping them to improve particular parts of their decision making that are the easiest to mathematically model. I expect to see rapid growth of DP-based "actor-critic" systems (i.e. decision advisory services) in coming years, and an increasing amount of research devoted to validating and measuring the impact of these systems on the welfare of individuals and the profitability of firms. Economists are enduring a period of time where their level of credibility and influence in high level policy making is at historically low point. Some of this may be self-inflicted by a somewhat clueless assumption that individuals and firms do just fine, since they are unboundedly rational and act "as if" they solved DP problems, and thus need no help from us. Our path to redemption may start by trying to be more in touch with reality, and facing up to and trying to solve the sorts of normative decision problems that Herbert Simon challenged us to confront in his 1978 Nobel Prize lecture. We need to show that our arsenal of mathematical tools and skills, including DP and econometrics, have numerous real world applications and provide credible demonstrations that these tools really can improve individual and firm decision making.

References

Adda, J., & Cooper, R. (2003). *Dynamic economics quantitative methods applications*. MIT Press.

Aguirregabiria, V., & Mira, P. (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, *156*(1), 38–67.

Anderson, & Kimes. (2011). Revenue management for enhanced profitability an introduction for hotel owners and asset managers. In M. C. Sturman, J. B. Corgel, & R. Verma (Eds.), *The Cornell School of Hotel Administraton on hospitality* (pp. 192–206). John Wiley & Sons, Inc.

Arrow, K., Harris, T., & Marshak, J. (1951). Optimal inventory policy. *Econometrica*, *19*, 250–272.

Barron, A. (1994). Approximation and estimation bounds for neural networks. *Machine Learning*, *14*(1), 115–133.

Barto, A., Bradtke, S., & Singh, S. (1995). Learning to act using real time dynamic programming. *Artificial Intelligence*, *72*, 81–138.

Barto, A., & Dietterich, T. (2004). Reinforcement learning and its relation to supervised learning. In J. Si, A. Barto, W. Powell, & D. Wunsch (Eds.), *Learning and approximate dynamic programming: Scaling up to the real world* (pp. 46–63). Wiley Interscience.

Bellman, R. (1984). *Eye of the hurricane*. World Scientific Publishing Company, Singapore.

Bertsekas, D. (1982). Distributed dynamic programming. *IEEE Transactions on Automatic Control*, *27*, 610-616.

Bertsekas, D. (2017). *Dynamic programming and optimal control volume 1*. Belmont, Massachusetts, Athena Scientific.

Bertsekas, D., & Tsitsiklis, J. (1989). *Parallel and disrributed computation: Numerical methods*. Prentice Hall, Englewood Cliffs, New Jersey.

Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific, Belmont, Massachusetts.

Bilonis, I., & Scheidegger, S. (2017). Machine learning for high-dimensional dynamic stochastic economies. *SSRN working paper*.

Blackwell, D. (1965). Discounted dynamic programming. *Annals of Mathematical Statistics*, *36*(1), 226–235.

Bloise, G., & Vailakis, Y. (2018). Convex dynamic programming with (bounded) recursive utility. *Journal of Economic Theory*, *173*, 118-141.

Brown, G. (1951). Iterative solutions of games by fictitious play. In T. C. Koopmans (Ed.), *Activity analysis of production and allocation.* Wiley, New York.

Brumm, J., & Scheidegger, S. (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica*, *85*(5), 1575–1612.

Camerer, C. (2003). *Behavioral game theory*. Princeton University Press.

Campbell, J., & Shiller, R. (1988). Stock prices, earnings and expected dividends. *The Journal of Finance*, *43*(3).

Carbone, E., & Duffy, J. (2014). Lifecycle consumption plans, social learning and external habits: Experimental evidence. *Journal of Economic Behavior and Organization*, *106*, 413–427.

Case, A., & Deaton, A. (2015). Rising morbidity and mortality in midlife among white non-hispanic americans in the 21st century. *Proceedings of the National Academy of Sciences*, *112*(49), 15078–15083.

Chan, T., & Hamilton, B. (2006). Learning, private information, and the economic evaluation of randomized experiments. *Journal of Political Economy*, *114*(64), 997–1040.

Cho, S., Lee, G., Rust, J., & Yu, M. (2018). Optimal dynamic hotel pricing. *working paper, Department of Economics, Georgetown University*.

Cho, S., & Rust, J. (2010). The flat rental puzzle. *Review of Economic Studies*, *77*(2), 560–594.

Chow, C., & Tsitsiklis, J. (1989). The complexity of dynamic programming. *Journal of Complexity*, *5*(4), 466–488.

Chow, G. (1976). *Analysis and control of dynamic economic systems*. Wiley, New York.

Crawford, G., & Shum, M. (2005). Uncertainty and learning in pharmaceutical demand. *Econometrica*, *73*(4), 1137–1173.

Dellavigna, S. (2018). Structural behavioral economics. In D. Bernheim, S. Dellavigna, & D. Laibson (Eds.), *Handbook of behavioral economics, volume 1.* Elsvier, North Holland.

Dreyfus, S. (2002). Richard bellman on the birth of dynamic programming. *Operations Research*, *50*(1), 48–51.

Eagleman, D. (2011). *Incognito: The secret lives of the brain*. Pantheon Books, Random House, New York.

Eckstein, Z., & Wolpin, K. (1989). The specification and estimation of dynamic discrete choice models. *Journal of Human Resources*, *24*, 562–598.

Ellis, C., Munnell, A., & Eschtruth, A. (2014). *Falling short: The coming retirement crisis and what to do about it*. Oxford University Press.

Forbes. (2018). What not to wear: how algorithms are taking uncertainty out of fashion. *Forbes*.

Frederick, S., Loewenstein, G., & O'Donoghue, T. (2002). Time discounting and time preference: A critical review. *Journal of Economic Literature*, *40*, 351–401.

Fung, K. (2014). When to hold out for a lower airfare. *FiveThirtEight*, *April 20*.

Gallego, G., & van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, *40*(8), 999–1020.

Gittins, J. (1989). *Multi-armed bandit allocation indices*. Wiley.

Griffiths, T., & Tenenbaum, J. (2009). Theory-based causal induction. *Psychological Review*, *116*(4), 661–716.

Gupta, S., & Rust, J. (2018). A simple theory of when and when firms go public. *working paper, Department of Economics, Georgetown University*.

Hall, G., & Rust, J. (2003). Middlemen versus market makers: A theory of competitive exchange. *Journal of Political Economy*, *111*(2), 353–403.

Hall, G., & Rust, J. (2007). The (s,s) policy is an optimal trading strategy in a class of commodity price speculation problems. *Economic Theory*, *30*, 515–538.

Hall, G., & Rust, J. (2018). *Econometric methods for endogenously sampled time series: The case of commodity price speculation in the steel market* (Tech. Rep.). Georgetown University.

Hamilton, B., Hincapie, A., Miller, R., & Papageorge, N. (2018). Learning, private information, and the economic evaluation of randomized experiments. *working paper, Carnegie Mellon University*.

Heckman, J., & Singer, B. (2017). Abducting economics. *American Economic Review*, *107*(5), 298–302.

Holt, C., Modigliani, F., Muth, J., & Simon, H. (1960). *Planning production, inventories and work force*. Prentice-Hall, Englewood Cliffs, New Jersey.

Howard, R. (1960). *Dynamic programming and markov processes*. MIT Press.

Hutchinson, J., & Meyer, R. J. (1994). Dynamic decision making: Optimal policies and actual behavior in sequential choice problems. *Marketing Letters*, *5*(4), 369–382.

Institute, I. R. (2018). Boomer expectations for retirement 2018. *Insured Retirement Institute*, *April*.

Iskhakov, F., Rust, J., & Schjerning, B. (2015). Recursive lexicographical search: Finding all markov perfect equilibria of finite state directional dynamic games. *Review of Economic Studies*, *83*(2), 658–703.

Iskhakov, F., Rust, J., & Schjerning, B. (2018). The dynamics of bertrand price competition with cost-reducing investments. *International Economic Review*.

James, K., & Tracy, D. (2018). Experimental tests of the life cycle model with investment in health capital. *working paper, Economic Science Institute, Chapman University*.

Johnson, S., Kotlikoff, L., & Samuelson, W. (2001). Can people compute? an experimental test of the life cycle consumption model. In L. Kotlikoff (Ed.), *Essays on saving, bequests, altruism, and life-cycle planning* (pp. 335–385). MIT Press, Cambridge, Massachusetts.

Judd, K. (1998). *Numerical methods in economics*. MIT Press.

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Karp, R. (1972). Reducibility among combinatorial problems. In R. Miller & J. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–104). Plenum Press, New York.

Kumar, P. R. (1989). A survey of some results in stochastic adaptive control. *SIAM Journal of Control and Optimization*, *23*, 329–380.

Leary, M., & Michaely, R. (2011). Determinants of dividend smoothing: empirical evidence. *Review of Financial Studies*, *24*(10), 3197–3249.

Lewis, M. (2003). *Moneyball: The art of winning and unfair game*. W.W. Norton & Company.

Lusari, A., & Mitchell, O. (2014). The economic importance of financial literacy: Theory and evidence. *Journal of Economic Literature*, *52*(1), 5–44.

Magnac, T., & Thesmar, D. (2002). Identifying discrete decision processes. *Econometrica*, *70*(2), 810–816.

Mak, T., Cheung, P., Lam, K., & Luk, W. (2011). Adaptive routing in network-on-chips using a dynamic-programming network. *IEEE Transactions on Industrial Electronics*, *58*(8), 3701–3716.

Markets, & Markets. (2016). *Revenue management market by solutions (risk management, pricing and revenue forecast management, revenue analytics, revenue leakage detection, channel revenue management) by services (professional, managed) by deployment mode - global forecast to 2020* (Tech. Rep.). Author.

Maskin, E., & Tirole, J. (2001). Markov perect equilibrium i: Observable actions. *Journal of Economic Theory*, *100*(1), 191–219.

Massé, P. (1944). Application des probabilités en chain á l'hydrologie statistique et au jeu des réservoirs. *Soc. Stat. Paris*, *86*(9-10), 204–219.

McAfee, P., & te Veld, V. (2008). Dynamic pricing with constant demand elasticity. *Production and Operations Management*, *17*(4), 432–438.

McClelland, J., & Rust, J. (2018). Strategic timing of investment over the business cycle: Machine replacement in the us rental industry. *Journal of Economics and Statistics*, *238*(3-4), 295–352.

Miller, J., Palmer, R., & Rust, J. (1993). Behavior of trading automata in a computerized double auction market. In D. Friedman & J. Rust (Eds.), *The double auction market: Theory, institutions, and laboratory evidence.* Addison Wesley, Redwood City, CA.

Misra, S., & Nair, H. (2011). A structural model of sales-force compensation dynamics: Estimation and field implementation. *Quantitative Marketing and Economics*, *9*, 211–257.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., ... Hassabis, D. (2015). Human-level conrol through deep reinforcement learning. *Nature*, *518*, 529–533.

Myerson, R. (1981). Optimal auction design. *Mathematics of Operations Research*, *6*(1), 58–73.

Phillips, R. L. (2005). *Pricing and revenue optimization*. Stanford University Press.

Pollatschek, M., & Avi-Itzhak, B. (1969). Algorithms for stochastic games with geometrical interpretation. *Management Science*, *15*, 399–413.

Powell, W. (2010). *Approximate dynamic programming solving the curses of dimensionality*. Wiley, New York.

Powell, W., Bouzaiene-Ayari, B., Lawrence, C., Cheng, C., Das, S., & Fiorillo, R. (2014). Locomotive planning at norfolk southern: An optimizing simulator using approximate dynamic programming. *Interfaces*, *44*(6), 567–578.

Puterman, M. (2005). *Markov decision processes: Discrete stochastic dynamic programming*. New York: Wiley Series in Probability and Statistics.

Robbins, H., & Munro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*(3), 400–425.

Romer, D. (2006). Do firms maximize? evidence from professional football. *Journal of Political Economy*, *114*(2), 340–364.

Rust, J. (1987). Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, *55*(5), 993–1033.

Rust, J. (1992). *Do people behave according to bellman's principle of optimality?* Hoover Institution, Stanford University, 69 pages.

Rust, J. (1994). Structural estimation of markov decision processes. In R. Engel & D. McFadden (Eds.), *Handbook of econometrics, volume iv.* Elsevier, North Holland.

Rust, J. (1996). Numerical dynamic programming in economics. In H. Amman, D. Kendrick, & J. Rust (Eds.), *Handbook of computational economics.* Elsevier, North Holland.

Rust, J. (1997). Using randomization to break the curse of dimensionality. *Econometrica*, *65*(3), 487–516.

Rust, J. (2008). Dynamic programming. *New Palgrave Dictionary of Economics*.

Rust, J. (2014). The limits of inference *with* theory: A review of wolpin (2013). *Journal of Economic Literature*, *52*(3), 820–850.

Rust, J. (2017). Dynamic programming, numerical. *Wiley StatsRef*, 1–10.

Scholz, K., Sheshadri, A., & Khitatrakun, S. (2006). Are americans saving 'optimally' for retirement? *Journal of Political Economy*, *114*(4), 607–643.

Silver, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv*, *1712*(010815v1), 1–19.

Silver, D., Schrittewieser, J., Simonyan, K., Antonoglu, I., Huang, A., Guez, A., … Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, *550*, 354–358.

Simon, H. (1992). Rational decision-making in business organizations. In A. Lindbeck (Ed.), (pp. 343–371). World Scientific Publishing, Company.

Skinner, J. (2007). Are you sure you're saving enough for retirement? *Journal of Economic Perspectives*, *21*(3), 59–80.

Stokey, N., & Lucas, R. (1989). *Recursive methods in economic dynamics*. Harvard University Press.

Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44.

Thaler, R., & Sunstein, C. (2008). *Nudge*. Yale University Press.

Traub, J., & Werschulz, A. G. (1998). *Complexity and information*. Academia Nazionale Dei Lincei.

Tsitsiklis, J. (1995). Asynchronous stochastic approximation and q-learning. *Machine Learning*, *16*, 185–202.

Wald, A. (1947). Foundations of a general theory of statistical decision functions. *Econometrica*, *15*, 279–313.

Watkins, C. (1989). Learning from delayed rewards. *Ph.D. thesis, Cambridge University, Cambridge, UK*.

Wiener, N. (1948). *Cybernetics: Or control and communication in the animal and the machine*. Hermann and Cie, Paris.

Williams, K. R. (2018). Dynamic airline pricing and seat availability. *Cowles Foundation discussion paper 3003U*.

Zhang, K., Wu, T., Chen, S., Cai, L., & Peng, C. (2017). A new energy efficient vm scheduling algorithm for cloud computing based on dynamic programming. *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, 249–254.