Sampled fictitious play for multi-action stochastic dynamic programs

Archis Ghate[*] Shih-Fen Cheng Stephen Baumert Daniel Reaume
Dushyant Sharma Robert L. Smith

February 2, 2013

Abstract

We introduce a class of finite-horizon dynamic optimization problems that we call multi-action stochastic dynamic programs (DPs). Their distinguishing feature is that the decision in each state is a multi-dimensional vector. These problems can in principle be solved using Bellman's backward recursion. However, complexity of this procedure grows exponentially in the dimension of the decision vectors. This is called the curse of action-space dimensionality. To overcome this computational challenge, we propose an approximation algorithm rooted in the game theoretic paradigm of Sampled Fictitious Play (SFP). SFP solves a sequence of DPs with a one-dimensional action-space, which are exponentially smaller than the original multi-action stochastic DP. In particular, the computational effort in a fixed number of SFP iterations is linear in the dimension of the decision vectors. We show that the sequence of SFP iterates converges to a local optimum, and present a numerical case study in manufacturing where SFP is able to find solutions with objective values within 1% of the optimal objective value hundreds of times faster than the time taken by backward recursion. In this case study, SFP solutions are also better by a statistically significant margin than those found by a one-step lookahead heuristic.

[*]Corresponding author. Industrial and Systems Engineering, Box 352650, The University of Washington, Seattle, WA 98195. Email: archis@uw.edu.

1 Introduction

Consider the following class of finite-horizon stochastic dynamic programs (DPs). At the beginning of periods $t = 1, 2, \ldots, T$, a decision-maker observes the state $s_t \in S_t$ of a system, where S_t are non-empty and finite sets with cardinality $|S_t|$. He then chooses an n-dimensional decision vector $a = (a_1, a_2, \ldots, a_n)$, for some $n > 1$, where $a_i \in A_i(s_t)$, for $i = 1, 2, \ldots, n$. Here, $A_i(s_t)$ are non-empty and finite sets that may depend on s_t; we define $A(s_t)$ as the Cartesian product $A_1(s_t) \times A_2(s_t) \ldots \times A_n(s_t)$. The system then transitions into state $s_{t+1} \in S_{t+1}$ with probability $p_t(s_{t+1}; s_t, a)$, and accrues expected reward $r_t(s_t, a)$. The goal is to choose decisions that maximize the total T-period expected reward starting in initial state s_1 (thus S_1 is a singleton). We term these problems unconstrained multi-action stochastic DPs because (i) the decision vectors are multi-dimensional, and (ii) the decision vector in each state belongs to a Cartesian product of finite sets and thus do not have cross-component linking constraints. More generally, the reader can also imagine *constrained* multi-action stochastic DPs, where the decision vectors must satisfy cross-component coupling constraints and hence belong to a subset $F(s_t)$ of the Cartesian product $A(s_t)$. We implicitly assume that sets S_t do not depend on n, and in particular that they are "small." Thus, as we shall see in Section 2, the dimension n of the decision vectors is what makes the problem difficult, and this is the type of problems for which our methodology is particularly suitable. Multi-action stochastic DPs provide a natural modeling framework for several sequential decision problems. A few prototypical examples are discussed next.

1.1 Applications of multi-action stochastic dynamic programs

First consider a basic dynamic resource allocation problem where a known set of heterogeneous activities can be performed in each time-period. A random additional amount of one resource becomes available at the beginning of every time-period. The resource is necessary for performing any activity. Each activity can be performed at a finite number of different levels. The resource consumed by an activity as well as the expected reward obtained from that activity depend on the level at which the activity is performed. The decision maker chooses the activity levels at the beginning of every period after observing the available resource; the resource remaining after allocation can be carried forward to the next period perhaps by incurring an inventory holding cost.

Reward functions and cost functions may vary across time-periods. The goal is to maximize the total finite-horizon expected net reward. The economic tradeoff is between immediately spending the resource on activities that appear valuable versus saving some of the resource, incurring holding costs in some cases, with the hope that it can be utilized for higher rewards in the future.

This dynamic resource allocation problem is a constrained multi-action stochastic DP and its detailed formulation is given in Appendix A. The dimension n of the decision vectors equals the total number of activity types. The state corresponds to the amount of resource available at the beginning of a period. The set of feasible decision vectors is defined by the constraint that the total amount of resource consumed by all activities cannot exceed the amount of resource that is available. Through a simple transformation that we describe in Appendix A, this constrained multi-action stochastic DP can in fact be converted to an unconstrained multi-action stochastic DP.

An appropriately modified version of the dynamic resource allocation problem, where the resource corresponds to the wealth of an investor and the activities correspond to different financial assets, leads to a well-known class of dynamic portfolio optimization problems [13] that can be formulated as multi-action stochastic DPs.

The following dynamic pricing problem [19] is also closely related to the dynamic resource allocation problem and can be modeled as a multi-action stochastic DP. A firm is endowed with a fixed initial quantity of a resource that can be used for providing different services from a fixed set of possible services. The resource cannot be replenished over the problem horizon. Each service offered consumes a fixed quantity of the resource. In each period, after observing the amount of available resource, the firm decides which services to offer, and what the corresponding prices should be. These prices induce a random demand for the services thus generating a random revenue. The goal is to maximize the total expected revenue over the problem horizon.

Another variation where the resource corresponds to a raw material such as crude oil, and the activities correspond to different end products such as different refined fractions [15], leads to a dynamic resource procurement and product-mix problem. In that case, the raw material does not randomly arrive but must be purchased at a price that itself may be assumed to follow a suitable stochastic process hence requiring an appropriate modification of the state definition. Moreover, the amount of resource purchased also becomes a decision variable. This problem can also be formulated as a multi-action stochastic DP.

The dynamic pricing and the dynamic resource procurement and product-mix problems possess a structural feature that often arises in manufacturing and service operations management. In particular, in these problems, different kinds of decisions need to be made in each time-period: in the dynamic pricing problem, service types as well as prices need to be chosen; in the dynamic resource procurement and product-mix problem, a resource procurement strategy as well as product types must be selected. Such heterogeneous decisions were traditionally optimized separately in a hierarchical manner. However, the more recent trend in the operations management literature is toward an integrated view where decisions at strategic, tactical and operational decisions are optimized in a single optimization model. Examples include integrated decision models for simultaneous pricing and production/inventory decisions [6, 9] and jointly optimizing dynamic pricing and sequencing policies for multi-product, single-server queuing systems [18]. Such integrated decision problems can be formulated as multi-action stochastic DPs, and in fact, the manufacturing application presented in Section 5 exhibits this feature.

We next describe the key computational challenge in solving multi-action stochastic DPs.

2 Computational difficulties in finding an exact solution

For simplicity of exposition, we focus in this and the next two sections on unconstrained multi-action stochastic DPs. As demonstrated in Appendix A for the special case of dynamic resource allocation problems and also in our manufacturing case study in Section 5, a constrained mutli-action stochastic DP can be converted into an equivalent, unconstrained one by mapping infeasible solutions into feasible solutions. Consequently, the ideas developed here continue to apply to constrained multi-action stochastic DPs as well. (As long as one feasible solution to the constrained multi-action stochastic DP is known, such a transformation is always possible by mapping all infeasible solutions to this single feasible solution.)

First, we recall standard terminology from DP (see [1, 23, 24, 26]). A (Markovian deterministic) policy $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ is a mapping that assigns n-dimensional decision vector $\pi_t(s_t) = (\pi_{1t}(s_t), \pi_{2t}(s_t), \ldots, \pi_{nt}(s_t)) \in A(s_t)$ to state $s_t \in S_t$ in period t. The set of all such policies is denoted by Π. Let $V_\pi(s_1)$ denote the total T-period expected reward obtained by implementing policy π starting in initial state s_1. It is given by

$$V_\pi(s_1) \triangleq E_\pi\left[\sum_{t=1}^{T} r_t(Z_t, \pi_t(Z_t))\bigg| Z_1 = s_1\right], \tag{1}$$

where $Z_2, \ldots, Z_T$ are the random states occupied beginning periods $2, \ldots, T$, respectively, given that the initial state Z_1 is s_1 and policy π is implemented.

There exists a policy in Π that maximizes the total T-period expected reward among all decision rules. Thus a policy is said to be optimal if it maximizes $V_\pi(s_1)$ over all $\pi \in \Pi$. The decision-maker's problem is hence given by

$$V_1(s_1) \triangleq \max_{\pi \in \Pi} V_\pi(s_1). \tag{2}$$

More generally, let $V_t(s_t)$ denote the maximum total expected reward in periods $t, \ldots, T$ when starting period t in state s_t. Problem (2) can in principle be solved using Bellman's equations

$$V_t(s_t) = \max_{a \in A(s_t)} \left\{ r_t(s_t, a) + \sum_{s_{t+1} \in S_{t+1}} p_t(s_{t+1}; s_t, a) V_{t+1}(s_{t+1}) \right\} \ \forall s_t \in S_t, \tag{3}$$

in the order $t = T, \ldots, 1$, where $V_{T+1}(s_{T+1}) = 0$ for all $s_{T+1} \in S_{T+1}$. Decisions that achieve the above maxima define an optimal policy in Π.

If cardinality $|A_i(s_t)| = M$ for each i, then cardinality $|A(s_t)| = M^n$. Thus the computational effort required to perform one maximization in (3) grows ***exponentially with*** n . This is called the curse of action-space dimensionality [23] and it makes exact solution of a multi-action stochastic DP intractable. We propose an approximation algorithm rooted in a game theoretic best response paradigm called Sampled Fictitious Play (SFP) [16]. As we shall see, n DPs, each with a one-dimensional action-space, are solved in each iteration of SFP and hence the computational effort in one iteration grows ***linearly with*** n. We will prove that SFP converges to locally optimal solutions of the multi-action stochastic DP, and will demonstrate through numerical results that SFP can obtain high-quality solutions very quickly. In order to make some of these ideas rigorous, we need to view the original multi-action stochastic DP as an equivalent *game of common interests* between n players as described next.

3 An equivalent game of common interests

We view an unconstrained multi-action stochastic DP as a game of common interests in which player i corresponds to the ith component of the decision vector. A pure strategy $\psi_i = (\psi_{i1}, \ldots, \psi_{iT})$ for player i is a mapping that prescribes decisions $\psi_{it}(s_t) \in A_i(s_t)$ in every state $s_t \in S_t$ in every period t. Let Ψ_i denote the non-empty and finite set of all such pure strategies for player i. A pure joint strategy for all players is given by $\psi = (\psi_1, \psi_2, \ldots, \psi_n)$. The set of all such pure joint strategies is denoted by $\Psi = \Psi_1 \times \Psi_2 \times \ldots \times \Psi_n$. Each $\psi \in \Psi$ defines a unique policy $\pi \in \Pi$ that prescribes the decision vector $\pi_t(s_t) = (\psi_{1t}(s_t), \psi_{2t}(s_t), \ldots, \psi_{nt}(s_t))$ in state s_t in period t. We express this one-to-one relationship between pure joint strategies in Ψ and policies in Π by using an operator $O : \Psi \to \Pi$. Note that for every policy $\pi \in \Pi$, there exists a (unique) pure joint strategy $\psi \in \Psi$ such that $O(\psi) = \pi$. In particular, this ψ is precisely the pure joint strategy in which player i's pure strategy prescribes decision $\pi_{it}(s_t)$ in state s_t in period t, that is, $\psi_{it}(st) = \pi_{it}(s_t)$. That is, O is onto. The identical payoff $U(\psi)$ received by each player on playing pure joint strategy ψ is defined as the total T-period expected reward obtained by implementing policy $O(\psi) \in \Pi$. Mathematically,

$$U(\psi) \triangleq V_{O(\psi)}(s_1), \tag{4}$$

for all $\psi \in \Psi$. The players then solve a game of common interests given by

$$\max_{\psi \in \Psi} U(\psi). \tag{5}$$

This game has previously been called a common interest stochastic game, or a team Markov game (see, for example, [2]). We have

Lemma 3.1. *If $\psi \in \Psi$ is optimal to (5), then $O(\psi)$ is optimal to (2).*

Proof. We have $U(\psi) \geq U(\xi)$ for all $\xi \in \Psi$. Thus (4) implies that $V_{O(\psi)}(s_1) \geq V_{O(\xi)}(s_1)$ for all $\xi \in \Psi$. This proves the claim because O is onto as explained in the first paragraph in this section. $\square$

We now review basic concepts from game theory (see [10]) and establish that Nash equilibria of (5) are locally optimal for (2). We use ψ_{-i} to denote pure strategies $(\psi_1, \ldots, \psi_{i-1}, \psi_{i+1}, \ldots, \psi_n)$

of all players except player i (subscript $-i$ is interpreted similarly everywhere). For $\mu_i \in \Psi_i$ and $\psi \in \Psi$, (μ_i, ψ_{-i}) denotes the pure joint strategy in which player i plays μ_i and the others play ψ_{-i}.

Definition 3.2. *A pure joint strategy $\psi \in \Psi$ is a Nash equilibrium for (5) if, for each $i = 1, 2, \ldots, n$,*

$$U(\psi) \geq U(\mu_i, \psi_{-i}), \ \forall \mu_i \in \Psi_i. \tag{6}$$

That is, if no player can increase the payoff to all players by unilaterally altering its own pure strategy.

An optimal pure joint strategy for (5) is a Nash equilibrium for (5). Conversely, a pure joint strategy Nash equilibrium for (5) can be seen as a local optimum for (5) because moving away from a Nash equilibrium along any of the "coordinate directions" does not improve the common objective. Formally, let $\mathcal{N}(\psi) \triangleq \{(\mu_i, \psi_{-i}) : \mu_i \in \Psi_i, \text{ for some } i\}$ be a neighborhood of $\psi \in \Psi$, and note that (6) can be written as $U(\psi) \geq U(\mu)$ for all $\mu \in \mathcal{N}(\psi)$.

Lemma 3.3. *If ψ is a pure joint strategy Nash equilibrium for (5), then $O(\psi)$ is locally optimal for (2) in the sense that $V_{O(\psi)}(s_1) \geq V_{O(\mu)}(s_1)$ for all $\mu \in \mathcal{N}(\psi)$.*

Proof. We have, $V_{O(\psi)}(s_1) = U(\psi) \geq U(\mu) = V_{O(\mu)}(s_1)$ for all $\mu \in N(\psi)$, by (4) and since ψ is a Nash equilibrium. $\square$

Thus, although the concept of local optimality has not been used in the published approximate dynamic programming literature to the best of our knowledge (see, however, a recent doctoral dissertation on near-optimality in sequential decision problems [27]), in the context of multi-action stochastic DPs it is identical to the notion of local optimality in static optimization — a solution is locally optimal if it has the best objective value in its neighborhood.

We define the set Δ_i of all mixed strategies for player i as $\Delta_i \triangleq \left\{ f_i : \Psi_i \to [0,1] : \sum_{\psi_i \in \Psi_i} f_i(\psi_i) = 1 \right\}$. Each $f_i \in \Delta_i$ can be viewed as an assignment of probabilities, or beliefs, to the elements of Ψ_i. We identify pure strategies $\psi_i \in \Psi_i$ as members of Δ_i that assign probability 1 to ψ_i. We define $\Delta \triangleq \Delta_1 \times \Delta_2 \times \cdots \Delta_n$ and an $f = (f_1, f_2, \ldots, f_n) \in \Delta$ is called a mixed joint strategy. Note that the set Δ can be viewed as belonging to a Euclidean space. We extend the domain of the payoff function U such that for any mixed joint strategy $f \in \Delta$, we have,

$$U(f) \triangleq \sum_{\psi\in\Psi} U(\psi)f(\psi) = \sum_{\psi\in\Psi} U(\psi_1, \psi_2, \dots, \psi_n) f_1(\psi_1) f_2(\psi_2) \cdots f_n(\psi_n). \tag{7}$$

We interpret this $U(f)$ as the expected payoff to all players when they independently choose pure strategies with probabilities defined by $f = (f_1, \dots, f_n)$. We now establish a relation between mixed joint strategies and a class of randomized decision rules for the multi-action stochastic DP.

We define a set Φ of randomized decision rules for the multi-action stochastic DP as $\Phi \triangleq \left\{\phi : \Pi \to [0,1] : \sum_{\pi\in\Pi} \phi(\pi) = 1\right\}$. That is, ϕ chooses Markovian deterministic policies π with probabilities $\phi(\pi)$. The total T-period expected reward accrued by a randomized decision rule $\phi \in \Phi$ is given by

$$V_\phi(s_1) \triangleq \sum_{\pi\in\Pi} V_\pi(s_1)\phi(\pi). \tag{8}$$

A mixed joint strategy $f \in \Delta$ defines a unique randomized decision rule denoted $\Theta(f) \in \Phi$, which chooses policies $O(\psi)$ with probabilities $f(\psi)$ for all $\psi \in \Psi$. This yields

$$U(f) = \sum_{\psi\in\Psi} U(\psi)f(\psi) = \sum_{\psi\in\Psi} V_{O(\psi)}(s_1)f(\psi) = \sum_{\pi\in\Pi} V_\pi(s_1)[\Theta(f)(\pi)] = V_{\Theta(f)}(s_1), \tag{9}$$

where the third equality holds because O is bijective and $f(\psi)$ is the same as $\Theta(f)(\pi)$ for $\pi = O(\psi)$.

Definition 3.4. *A mixed joint strategy $f \in \Delta$ is a Nash equilibrium, if for each player i,*

$$U(f) \geq U(g_i, f_{-i}) \ \ \forall g_i \in \Delta_i. \tag{10}$$

That is, if no player can increase the payoff to all players by altering its own mixed strategy.

For any $f \in \Delta$, its neighborhood $\mathcal{M}(f) \triangleq \{(g_i, f_{-i}) : g_i \in \Delta_i, \text{ for some } i\}$. Then (10) can be written as $U(f) \geq U(g)$ for all $g \in \mathcal{M}(f)$. This yields the following counterpart of Lemma 3.3.

Lemma 3.5. *If $f \in \Delta$ is a mixed joint strategy Nash equilibrium then the randomized decision rule $\Theta(f) \in \Phi$ is locally optimal for (2) in the sense that $V_{\Theta(f)}(s_1) \geq V_{\Theta(g)}(s_1)$ for all $g \in \mathcal{M}(f)$.*

Proof. By (9) and since f is a Nash equilibrium, $V_{\Theta(f)}(s_1) = U(f) \geq U(g) = V_{\Theta(g)}(s_1)$ $\forall g \in \mathcal{M}(f)$. □

Thus any efficient algorithm for finding Nash equilibria of (5) will efficiently discover locally optimal solutions of (2). We chose SFP [16], which in turn is based on fictitious play (FP) [3, 22, 25], for this purpose. This choice was motivated by (i) the high quality of solutions delivered by variants of FP in other discrete optimization problems [4, 7, 8, 12, 17]; (ii) previous evidence that the sampling process in SFP provides a good tradeoff between exploitation and exploration of the solution space [7, 8]; (iii) SFP's asymptotic convergence to equilibrium for games of common interests [16]; (iv) the fact, as we shall show, that SFP in every iteration solves n DPs, each with a one-dimensional action-space, instead of the original intractable n-dimensional DP; and (v) the ease with which SFP can be parallelized (see [7]) across the n components of the decision vectors, achieving a further n-fold reduction in wall-clock time needed.

4 Sampled fictitious play

Fictitious play [3, 25] was originally proposed to find Nash equilibria of two-player games and was later extended to n-player games [22]. In each iteration of FP, each player computes a pure strategy best response assuming that the other players will play according to the mixed strategy defined by their empirical frequency of past best responses. The empirical frequencies are then updated with these new best responses. For games of common interests, the sequence of empirical frequencies converges to equilibrium in the following sense. For every $\delta > 0$, there exists an iteration counter κ such that each empirical frequency in all iterations $k \geq \kappa$ is within a Euclidean distance δ from some (possibly mixed joint strategy) Nash equilibrium. See [16, 22] for a proof of this result. However, a straightforward implementation of FP is impractical when the sets of pure strategies are large. This is because each player's best response requires a computationally demanding calculation of that player's expected payoff given the other players' mixed strategies. SFP [16] overcomes this hurdle by approximating this expectation with a sample average (Steps 2(a), (b) below) and still retains convergence to equilibrium (Theorem 4.2 below) as in FP.

We now present SFP for the game of common interests from Section 3. We let f^k denote the empirical frequency of best responses of all players in the first k iterations. Formally, for any player i and any pure strategy $\psi_i \in \Psi_i$, $f_i^k(\psi_i)$ equals the proportion of iterations, out of k, in which pure strategy ψ_i was player i's best response. Thus f_i^k is a probability mass function over Ψ_i and hence belongs to Δ_i. Moreover, $f^k = (f_1^k, f_2^k, \ldots, f_n^k)$ belongs to Δ and hence is a mixed joint strategy.

Algorithm 4.1 (Sampled Fictitious Play).

1. **Initialization:** Set iteration $k = 1$ and select $\psi^0 \in \Psi = \Psi_1 \times \Psi_2 \times \ldots \times \Psi_n$ arbitrarily; set $f_i^0(\psi_i^0) = 1$ for all players i, and $f_i^0(\xi_i) = 0$ for all players i and all $\xi_i \in \Psi_i$ such that $\xi_i \neq \psi_i^0$.

2. **Iteration $k \geq 1$:**

   (a) **Sampling:** Select a sample size $N_k \geq 1$, and draw an iid pure strategy sample ψ_i^{jk}, for $j = 1, \ldots, N_k$, from the probability distributions f_i^{k-1}, for each player i.

   (b) **Best response:** For each player i, find a pure strategy best response to the sample ψ_{-i}^{jk}, for $j = 1, 2, \ldots, N_k$, drawn by the other players, by solving

   $$\psi_i^k \in \operatorname*{argmax}_{\psi_i \in \Psi_i} \left\{ \sum_{j=1}^{N_k} \frac{U(\psi_i, \psi_{-i}^{jk})}{N_k} \right\}. \tag{11}$$

   (c) **Belief update:** Update empirical frequencies of best responses for each player i by setting

   $$f_i^k(\psi_i^k) = \frac{1 + (k-1) f_i^{k-1}(\psi_i^k)}{k},$$
   $$f_i^k(\xi_i) = \frac{(k-1) f_i^{k-1}(\xi_i)}{k} \text{ for } \xi_i \in \Psi_i,\ \xi_i \neq \psi_i^k.$$

   Increment k by 1 and go to the next iteration.

The following theorem is proven in [16] for a general common interest game.

Theorem 4.2. *Suppose sample sizes $N_k = \lceil Ck^\beta \rceil$ for some $\beta > \frac{1}{2}$ and $C > 0$. Then, with probability 1, the sequence $\{f^k\}_{k=1}^\infty$ converges to equilibrium in the sense described above.*

Corollary 4.3. *Suppose sample sizes $N_k = \lceil Ck^\beta \rceil$ for some $\beta > \frac{1}{2}$ and $C > 0$. Then, with probability 1, the sequence $\{\Theta(f^k)\}_{k=1}^\infty$ of randomized decision rules converges to local optima for the multi-action stochastic DP.*

We now make a few observations that render this algorithm particularly easy to implement.

Efficient sampling: It is not necessary to explicitly maintain and update empirical frequencies f^k as this would be computationally demanding. Instead, each player only maintains and updates a history of its own best responses. The sampling in Step 2(a) for player i then simply involves sampling N_k iteration counters, that is, integers from the set $\{1, 2, \ldots, k-1\}$, with equal probability. The pure strategies stored at those locations in the history of best responses for player i then form the sample ψ_i^{jk}, for $j = 1, 2, \ldots, N_k$. This process is probabilistically equivalent to sampling policies according to f_i^{k-1}.

Samples of size one: For computational efficiency, we use $N_k = 1$ for all k rather than increasing N_k at a rate that guarantees convergence as per Theorem 4.2. SFP with samples of size one has been implemented previously on traffic signal control problems [7] and DPs that arise in stochastic inventory control problems [8], producing good solutions in both cases. We consider this modification to be in line with a similar practice in other stochastic search methods. For example, in Simulated Annealing [14], the temperature parameter should be decreased slowly with iterations in order to guarantee convergence to optimality, however, in practice, the temperature is reduced much faster. Samples of size one simplify the best response problem (11) significantly as described next.

Best response problem (11) is a DP with a one-dimensional action-space: Suppose $N_k = 1$, and let $\xi^k \in \Psi$ be the pure joint strategy sampled in iteration k. Then problem (11) for player i reduces to $\psi_i^k \in \operatorname{argmax}_{\psi_i \in \Psi_i} \; U(\psi_i, \xi_{-i}^k)$, or in other words, to

$$\psi_i^k \in \underset{\psi_i \in \Psi_i}{\operatorname{argmax}} \; V_{O(\psi_i, \xi_{-i}^k)}(s_1), \tag{12}$$

by (4). The above problem is a stochastic DP with a one-dimensional action-space since pure strategies of all players other than player i are fixed at ξ_{-i}^k. It can be solved using Bellman's equations

$$V_{it}^k(s_t) \triangleq \max_{a_i \in A_i(s_t)} \left\{ r_t(s_t, (a_i, \xi_{-it}^k(s_t))) + \sum_{s_{t+1} \in S_{t+1}} p_t(s_{t+1}; s_t, (a_i, \xi_{-it}^k(s_t))) V_{it+1}^k(s_{t+1}) \right\} \tag{13}$$

for all $s_t \in S_t$ in the order $t = T, T-1, \ldots, 1$, starting with $V^k_{iT+1}(s_{T+1}) = 0$ for all $s_{T+1} \in S_{T+1}$. The best response strategy ψ^k_i in problem (12) is given by decisions that achieve the above maxima, and the corresponding optimal value $V_{O(\psi^k_i, \xi^k_{-i})}(s_1)$ equals $V^k_{i1}(s_1)$. This discussion shows that each iteration of SFP solves n DPs, each with a one-dimensional action-space, instead of solving the original, intractable n-dimensional DP. Thus the effort expended in k iterations is proportional to $k \times M \times n$. The best response problem is also a DP when $N_k > 1$ but in the interest of brevity we defer that discussion to Appendix B.

Algorithm output: As is common in other stochastic search algorithms, we track the best value attained and the corresponding policy that achieves this value during algorithm execution. Specifically, the best value attained in iterations 1 through k is given by

$$V(k) \triangleq \max_{l=1,2,\ldots,k} \left\{ \max_{i=1,2,\ldots,n} V^l_{i1}(s_1) \right\}.$$

Suppose that $V(k)$ is achieved during the best response calculation for player $i^*(k)$ in iteration $1 \le l^*(k) \le k$. That is, $V(k) = V^{l^*(k)}_{i^*(k)1}(s_1)$. Also let $\xi^{l^*(k)}$ be the policy sampled in iteration $l^*(k)$, and recall that $\psi^{l^*(k)}_{i^*(k)}$ denotes player $i^*(k)$'s best response to $\xi^{l^*(k)}_{-i^*(k)}$ in this iteration. Then $\pi(k) = O\left(\psi^{l^*(k)}_{i^*(k)}, \xi^{l^*(k)}_{-i^*(k)}\right) \in \Pi$ is the best multi-action stochastic DP policy found by Algorithm 4.1 in the first k iterations. Thus, if the algorithm is terminated after K iterations, both $V(K)$ and $\pi(K)$ are available as algorithm output.

5 Application to integrated capital investment, dynamic pricing, production scheduling and sales decisions in manufacturing

The problem we study in this section was motivated by our industrial collaborators at the General Motors Advanced Manufacturing Laboratory at the University of Michigan. Consider a manufacturing firm that produces units of one product such as computers, cars, airplanes, or clothes. We model the manufacturing process over a finite planning horizon that is divided into periods of equal length. At the beginning of the horizon, a strategic decision is made as to what plant capacity to put in operation. At the beginning of each period, the unit price of the product is determined. This induces a random demand for the product during that period through a price-demand function. The number of units to be produced in that period is decided before observing the realized demand. We allow for unreliable equipment and hence the plant may not be actually able to produce these many units. Before observing the realized demand and the realized production, a goal is set as to how many units should be sold in that period. We assume that extra demand is lost, that is, back-ordering is not allowed. The units remaining at the end of a period can be carried to the next period as inventory and used to satisfy the demand in that and subsequent periods. Inventory left at the end of the planning horizon is assumed to have no value. The goal is to maximize profit where revenue is generated by selling the product and costs are incurred for building the plant, production, and storing inventory.

We use the following notation:

- The planning horizon T is divided into periods of equal duration denoted by the indices $t = 1, 2, \ldots, T$.
- The finite set of possible integer plant capacities is given by $\mathbf{M} \triangleq \{m_1, m_2, \ldots, m_{|\mathbf{M}|}\}$ where $0 < m_1 < m_2 < \ldots < m_{|\mathbf{M}|}$.
- The cost of building a plant with capacity $m \in \mathbf{M}$ is $C(m)$ incurred every period. Thus the total cost of building is $TC(m)$. It may seem more natural to incur the building cost as a single, lump-sum expense. We instead followed the equivalent amortized cost approach to prevent the building costs from overwhelmingly dominating the profit function in problems with short planning horizons. Note that any current cash flow can be easily converted into an equivalent present-value annuity having any desired term. Another reason to allocate capacity costs by time period is that some fixed costs (heating, cooling, taxes, security, etc.) are incurred each time period.
- The finite set of possible unit prices in each period is denoted by $\mathbf{P}$.
- We let $\mathbf{D} \triangleq \{D_1(\cdot), D_2(\cdot), \ldots, D_{|\mathbf{D}|}(\cdot)\}$ be the set of possible price-demand functions. The demand induced by a particular price $p \in \mathbf{P}$ is an integer $D_k(p)$ with probability Q_k independently of everything else.

- ρ is a random variable independent of everything else that equals the fraction of planned capacity actually available for production (reliability) in a period. The random variable takes value l_k from a finite set $\mathbf{L} \triangleq \{l_1, l_2, \dots, l_{|\mathbf{L}|}\}$ with probability P_{l_k}.

- The cost of production depends on the period t, the plant capacity m, the planned production level x, as well as the realized production $\hat{x}$, and is denoted by $c(t, m, x, \hat{x})$.

- The cost of carrying an inventory of i units at the beginning of period t is $h(t, i)$. The initial inventory is zero.

- The number of units planned to be sold at the end of a period is denoted z whereas the realized sales are denoted $\hat{z}$.

5.1 A multi-action stochastic DP formulation

Suppose capacity $m \in \mathbf{M}$ is chosen at the beginning of the planning horizon. For this fixed capacity value, the state of our manufacturing system beginning period $t = 1, 2, \dots, T$ is given by $(t, (i, m))$, where i is the inventory on hand and belongs to the set $\{0, 1, 2 \dots, (t-1)m\}$ as $(t-1)m$ is the maximum possible total production in periods $1, 2, \dots, t-1$. The set $F(t, (i, m))$ of jointly feasible pricing, production, and sales decision vectors (p, x, z) is defined by constraints

$$p \in \mathbf{P} \tag{14}$$

$$x \in \{0, 1, \dots, m\} \tag{15}$$

$$z \in \{0, 1, \dots, i + x\}. \tag{16}$$

For any nonnegative real number y, let $\lfloor y \rfloor$ denote the largest integer not bigger than y. After choosing decisions $(p, x, z) \in F(t, (i, m))$, a price-demand function $D_j(\cdot) \in \mathbf{D}$ and a reliability $l_k \in \mathbf{L}$ are realized, leading to actual production and sales

$$\hat{x} = \min\{x, \lfloor l_k m \rfloor\} \tag{17}$$

$$\hat{z} = \min\{z, i + \hat{x}, D_j(p)\} = \min\{z, i + \min\{x, \lfloor l_k m \rfloor\}, D_j(p)\}. \tag{18}$$

Thus the state of the system beginning the next period is $(t+1,(i+\hat{x}-\hat{z},m))$ with probability $P_{l_k}Q_j$. The expected profit is given by

$$r_t((i,m),(p,x,z)) \triangleq \left(\sum_{l_k\in\mathbf{L}}\sum_{D_j(\cdot)\in\mathbf{D}} P_{l_k}Q_j\Big[p\hat{z}-C(m)-c(t,m,x,\hat{x})-h(t+1,i+\hat{x}-\hat{z})\Big]\right).$$

A feasible policy π is a decision rule that assigns price, production and sales decisions $(p,x,z) = (\pi_1(t,(i,m)),\pi_2(t,(i,m)),\pi_3(t,(i,m))) \in F(t,(i,m))$ to every state $(t,(i,m))$. The set of all feasible policies is denoted Π. Then, the total T-period expected profit under policy $\pi\in\Pi$ is

$$V_\pi(0,m) = E_\pi\left[\sum_{t=1}^{T} r_t((I_t,m),(\pi_1(t,(I_t,m)),\pi_2(t,(I_t,m)),\pi_3(t,(I_t,m))))\middle| I_1=0,m\right],$$

where $I_2, I_3, \ldots$ is a sequence of inventory levels induced by the policy π given that the initial inventory I_1 is zero, and the plant capacity is fixed at $m\in\mathbf{M}$.

As a result, given plant capacity m, we wish to solve

$$V_1(0,m) \triangleq \max_{\pi\in\Pi}\ V_\pi(0,m). \tag{19}$$

This can in principle be achieved by solving Bellman's equations

$$V_t(i,m) \triangleq \max_{(p,x,z)\in F(t,(i,m))}\left(r_t((i,m),(p,x,z)) + \sum_{l_k\in\mathbf{L}}\sum_{D_j(\cdot)\in\mathbf{D}} P_{l_k}Q_jV_{t+1}(i+\hat{x}-\hat{z},m)\right) \tag{20}$$

in the order $t=T,T-1,\ldots,1$, where $V_{T+1}(i,m)=0$, and $\hat{x}$, $\hat{z}$ are given by (17) and (18), respectively. Once $V_1(0,m)$ is available from (20) for every $m\in\mathbf{M}$, the optimal capacity m^* can be obtained by enumeration as

$$m^* \triangleq \underset{m\in\mathbf{M}}{\operatorname{argmax}}\ V_1(0,m). \tag{21}$$

Thus, we first focus on solving (19), which is a multi-action stochastic DP with three-dimensional decision vectors. As our numerical experiments will show, exact solution of Bellman's equations (20) is computationally demanding. We instead apply our game theoretic approximation approach from Section 4, and demonstrate numerically that it finds near-optimal solutions very quickly.

However, first notice that (19) is not an unconstrained multi-action stochastic DP. This is because decisions x and z are linked through constraint (16) and hence the set $F(t,(i,m))$ of feasible decision vectors (p,x,z) does not have the Cartesian product structure that is essential in viewing a multi-action stochastic DP as a game. In the next section, we present a variable transformation that converts the constrained problem (19) into an unconstrained one and then describe the resulting game of common interests.

5.2 A variable transformation and a game of common interests

Instead of selecting the planned production directly, we set the planned production to a *fraction* of the plant capacity. Similarly, instead of selecting the planned sales directly, we set the planned sales to a *fraction* of the inventory that would be available if the planned production is realized. We do not alter the price decision.

We introduce some notation to make these ideas precise. Let $d_{\max}$ be the maximum demand that could be observed in a period and $d(i)$ be the maximum possible amount that we could sell in a period where the inventory on hand is i, that is,

$$d_{\max} \triangleq \max_{D_j(\cdot)\in\mathbf{D}} D_j(\min\{\mathbf{P}\}),$$

$$d(i) \triangleq \min\ \left\{d_{\max}, i + m_{|\mathbf{M}|}\right\}.$$

We define sets

$$B_1(t,(i,m)) \triangleq \mathbf{P}, \tag{22}$$

$$B_2(t,(i,m)) \triangleq \left\{0, \frac{1}{m_{|\mathbf{M}|}}, \frac{2}{m_{|\mathbf{M}|}}, \ldots, 1\right\}, \text{ and} \tag{23}$$

$$B_3(t,(i,m)) \triangleq \left\{0, \frac{1}{d(i)}, \frac{2}{d(i)}, \ldots, 1\right\}, \tag{24}$$

and let $B(t,(i,m)) = B_1(t,(i,m)) \times B_2(t,(i,m)) \times B_3(t,(i,m))$. This particular definition of sets B_2 and B_3 ensures that they have sufficient "resolution" such that the subsequent transformation of joint strategies to feasible policies is onto. Now consider a game between three players, indexed $1, 2, 3$, which we call the revenue management (RM) player, the production planning (PP) player, and the sales (S) player, respectively. Pure strategy ψ_1 of RM is a mapping that assigns prices

$\psi_1(t,(i,m)) \in B_1(t,(i,m))$ to all possible inventories i and all periods t (recall that plant capacity m is fixed while solving (19)). Similarly, pure strategy ψ_2 of PP is a mapping that assigns fractions $\psi_2(t,(i,m)) \in B_2(t,(i,m))$, and pure strategy ψ_3 of S is a mapping that assigns fractions $\psi_3(t,(i,m)) \in B_3(t,(i,m))$ to all i and t. The finite sets of pure strategies of RM, PP, and S are denoted by Ψ_1, Ψ_2, Ψ_3, respectively, and $\Psi \triangleq \Psi_1 \times \Psi_2 \times \Psi_3$ denotes the finite set of their pure joint strategies $\psi = (\psi_1, \psi_2, \psi_3)$.

Each pure strategy $\psi \in \Psi$ defines a unique policy $\pi = O(\psi) \in \Pi$, which assigns feasible decision vectors $(p, x, z) = (\pi_1(t,(i,m)), \pi_2(t,(i,m)), \pi_3(t,(i,m))) \in F(t,(i,m))$ to all i and t, where

$$p = \psi_1(t,(i,m)), \tag{25}$$

$$x = \lfloor m\psi_2(t,(i,m)) \rfloor, \tag{26}$$

$$z = \lfloor \psi_3(t,(i,m))\,(i + \lfloor m\psi_2(t,(i,m)) \rfloor) \rfloor. \tag{27}$$

After the demand function $D_j(\cdot) \in \mathbf{D}$ and reliability $l_k \in \mathbf{L}$ are realized, the production and sales are given by (17) and (18), respectively. Also note that corresponding to every $\pi \in \Pi$, there is at least one $\psi \in \Psi$, such that $O(\psi) = \pi$. That is, $O : \Psi \to \Pi$ is an onto transformation. Finally, on choosing pure joint strategy $\psi \in \Psi$, RM, PP, and S, receive a common payoff

$$U^m(\psi) \triangleq V_{O(\psi)}(0, m), \tag{28}$$

where the superscript m emphasizes that the plant capacity is fixed at m. This completes our definition of a game of common interests between RM, PP, and S, and Lemmas 3.1 and 3.3 hold. In fact, after defining mixed strategies for these three players, it is easy to see, by following the reasoning in Section 3, that Lemma 3.5 will hold as well. We do not repeat the details here for brevity. Now we are ready to describe our implementation of the SFP algorithm on this game of common interests.

5.3 Description of algorithm implementation

For each $m \in \mathbf{M}$, we implemented the simplified version of SFP with efficiently drawn samples of size one as described after Algorithm 4.1. Here we present the stochastic DPs that arose in the best response problems for the RM, PP, and S players. To be precise, sampled strategies and best response strategies should be indexed by this fixed value of m in the discussion below; however, we suppress this for simplicity.

Let ξ_i^k, for $i = 1, 2, 3$, denote the pure strategy sampled in iteration k by the RM, PP, and S players, respectively. Then the best response problems of these three players are respectively given by

$$\psi_1^k \in \underset{\psi_1 \in \Psi_1}{\operatorname{argmax}} \; U^m(\psi_1, \xi_2^k, \xi_3^k) \tag{29}$$

$$\psi_2^k \in \underset{\psi_2 \in \Psi_2}{\operatorname{argmax}} \; U^m(\xi_1^k, \psi_2, \xi_3^k) \tag{30}$$

$$\psi_3^k \in \underset{\psi_3 \in \Psi_3}{\operatorname{argmax}} \; U^m(\xi_1^k, \xi_2^k, \psi_3). \tag{31}$$

As described in Section 4, these three best response problems can be formulated as DPs. Following the notation in (13) we denote the optimal values in these three DPs by $V_{11}^k(0, m)$, $V_{21}^k(0, m)$, and $V_{31}^k(0, m)$, respectively. Suppose that for every $m \in \mathbf{M}$ the algorithm is terminated after K iterations. As is common in stochastic search algorithms, K can be chosen after some initial experimentation as roughly the iteration counter where the algorithm stopped making significant gains in objective value. We then propose the following method to obtain a plant capacity, and an integrated price, production and sales policy for the original multi-action stochastic DP. As described in Section 4, we tracked the best value found by our SFP algorithm. In particular, at the end of K iterations with capacity m, we have,

$$V(K; m) \triangleq \max_{k=1,2,\ldots,K} \left\{ V_{11}^k(0, m), V_{21}^k(0, m), V_{31}^k(0, m) \right\}. \tag{32}$$

Suppose that the best value $V(K; m)$ is achieved during player i's best response calculation in the kth iteration. That is, $V(K; m) = V_{i1}^k(0, m)$ for some $i \in \{1, 2, 3\}$ and some $1 \le k \le K$. Let $\pi^m = O(\psi_i^k, \xi_{-i}^k)$ be the multi-action stochastic DP policy in Π that achieved this value. The best plant capacity was then found using the approximate version

$$\bar{m} \in \underset{m \in \mathbf{M}}{\operatorname{argmax}} \; V(K; m) \tag{33}$$

of enumeration (21). The price, production, and sales policy is thus given by $\pi^{\bar{m}}$, and the corresponding objective value by $V(K;\bar{m})$.

5.4 Problem data and results of numerical experiments

5.4.1 Problem data

The model in Section 5.1 requires data on plant building costs, stochastic price-demand functions, production costs, inventory costs, and plant reliability. The general trend in some of these data as plotted in Figure 1 was obtained from our collaborators at the General Motors Advanced Manufacturing Laboratory at the University of Michigan. The actual numbers shown in this figure have been distorted for confidentiality. Specifically,

- The planning horizon was $T = 15$ periods, where each period corresponds to one year. We repeated our experiments with $T = 5$ and $T = 10$ and found no difference in SFP performance.
- There were 33 possible plant capacities ($|\mathbf{M}| = 33$), ranging from yearly capacity of 20,000 to 120,000 units.
- The set $\mathbf{P}$ included 813 prices in the range \$13,500 to \$27,500.
- We assumed a constant elasticity demand function [29] of the form $D(p) = e^{\alpha}p^{\beta}$. In order to introduce stochasticity, we parameterized demand functions $D_i(\cdot)$ in the set $\mathbf{D}$ by parameters α_i and β_i. In particular, we included three possible demand functions that indicate low demand, normal demand, and high demand. Fitting the data provided to us, this led to $(\alpha_i, \beta_i) \in \{(48.5573, -4.5076), (49.0478, -4.5076), (49.5383, -4.5076)\}$. Note that the lower the price, the higher the demand owing to the negative value of parameter β_i. In each period, the actual realized demand was chosen from one of these three functions with equal probability. We repeated our experiments with non-uniform probability mass functions and found no difference in SFP performance.
- The variable production cost (including both labor and material cost) was assumed to be linear in the number of vehicles produced. Specifically, the variable production cost per vehicle was assumed to decrease with increasing plant capacity due to economies of scale as shown in Figure 1 and stationary across time periods.

- The inventory holding cost per vehicle was assumed to be 20 percent of the unit variable production cost in that period.

- The plant reliability value ρ was assumed to be an element of the set

$$\mathbf{L} = \{0.6, 0.66, 0.7, 0.74, 0.8\}.$$

  One of these values was selected with equal probability in each period. Again, we repeated our numerical experiments with non-uniform probability mass functions and found no difference in SFP performance.

5.4.2 Numerical results

The numerical experiments were run on a computing cluster equipped with Intel Xeon 5400 series processors. The operating system was a variant of RedHat Enterprise Linux. The code was written in C++. The number of iterations K was set to 20 because the algorithm was empirically observed to stop making significant progress in objective value thereafter.

First, standard backward recursion was employed to solve Bellman's equations (20) for each of the thirty three values of $m \in \mathbf{M}$. These thirty three implementations of backward recursion were done *in parallel*. The time required for backward recursion depends on plant capacity m: the higher the capacity, higher the time required. Thus, the time taken to complete backward recursion for the largest capacity m=120,000 is reported in Table 1. Problem (21) can then be solved in negligible time to obtain an optimal plant capacity, an optimal pricing, production and sales policy, and the corresponding optimal objective value.

Since we have thirty three different capacities in $\mathbf{M}$, thirty three runs of our algorithm — each corresponding to one of these capacity values — were performed *in parallel*. Recall that the run corresponding to capacity $m \in \mathbf{M}$ yields $V(20; m)$ and π^m on termination as described at the end of Section 5.2. Problem (33) was then solved easily to obtain a capacity $\bar{m}$, a pricing, production and sales policy $\pi^{\bar{m}}$, and the corresponding objective value $V(20; \bar{m})$. In fact, sixty independent runs of this entire procedure were performed with randomly generated initial pure strategies and the approximate optimal values obtained at the end of these runs were averaged. The average

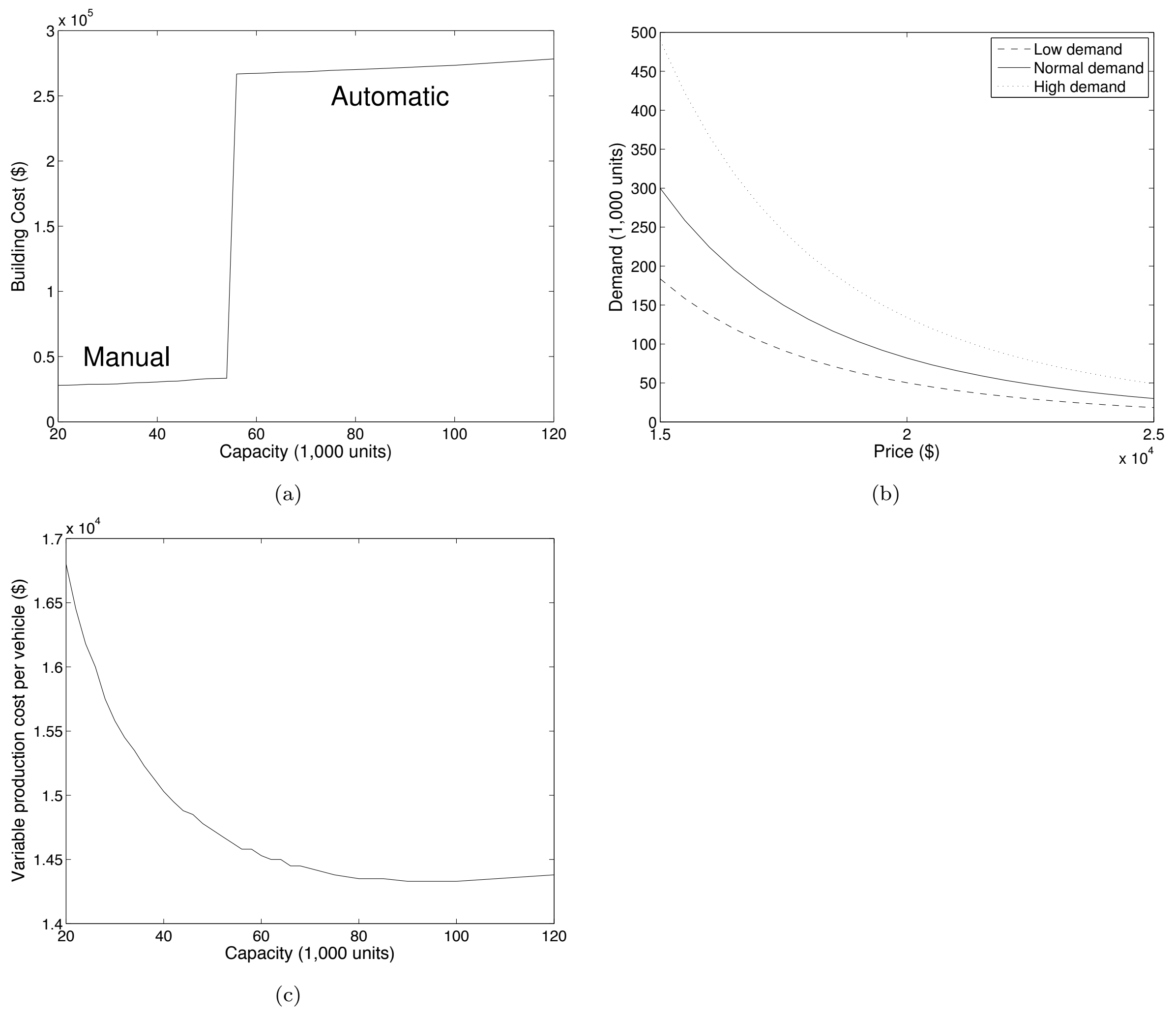

Figure 1: (a) Production line building cost, paid by period, as a function of capacity. The shape of the curve reflects the intuition that there is a capacity threshold below which a fully automated production line does not make sense; automation is essential above this threshold. (b) Demand as a function of price. (c) Variable production cost per vehicle as a function of capacity.

computation time required to complete the SFP procedure for the largest capacity m=120,000 is given in Table 1.

Algorithm	Execution time for m=120,000	Value estimate ratio (relative to the optimal value)
Backward Recursion	20.6 hours (1240 minutes)	1.0
SFP	5.4 minutes (averaged over 60 runs)	0.99

Table 1: Performance comparison between exact backward recursion and our SFP solver

From Table 1, we see that the SFP solver on average found objective values within 1% of the optimal objective value in only about 5 minutes. On the other hand, exact backward recursion takes about 20 hours of CPU time (about 227 times slower than our SFP solver). The progress of the SFP solver over twenty iterations is illustrated in Figure 2. The standard practice in sampling based algorithms for DP, and more generally, for global optimization, is to plot average algorithm performance over independent runs versus iterations (see for example [5, 28]). Following this approach, we track the best value observed by SFP as a fraction of the optimal value, and plot its average over sixty independent runs. We observe that the SFP solver makes most improvements during early iterations (the average fraction climbs to about 0.95 in just 5 iterations), and despite having large variation during the early iterations (due to randomly generated initial solutions), the ratio converges quickly later on. The range of fractions at the 20th iteration was only $[0.972, 0.999]$, with an average of 0.99. Thus even in the worst case (out of the sixty random instances of initial solutions), the SFP solver was within 3% of the optimal value.

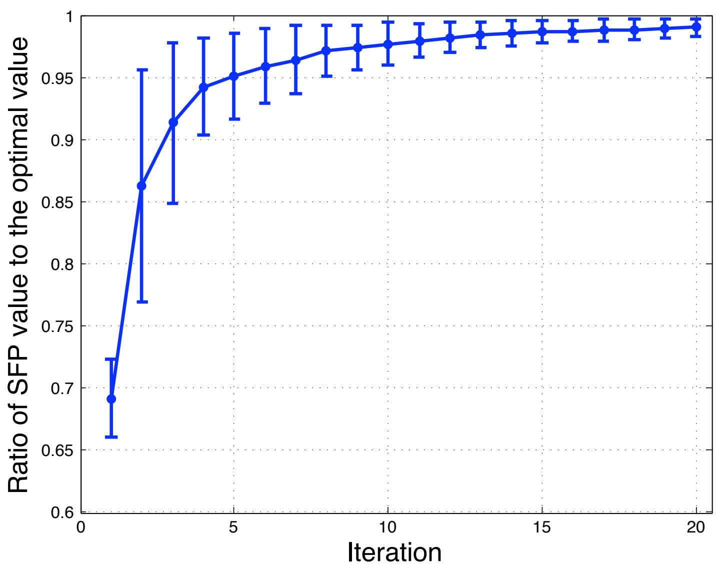

Figure 2: For each iteration, the average (over sixty independent runs with random initial solutions) ratio of best value found by SFP to the optimal value is plotted. The error bars surrounding each value correspond to +1 and -1 standard deviations.

We also compared SFP performance with a one-step lookahead heuristic (see Section 6.3 of [1]). In fact, we gave this heuristic a significant (unfair) advantage by implementing it for the *optimal* plant capacity m^* as found earlier by backward recursion. In state $(t, (i, m^*))$, this heuristic prescribes a decision that solves

$$\max_{(p,x,z)\in F(t,(i,m^*))}\left(r_t((i,m^*),(p,x,z))+\sum_{l_k\in\mathbf{L}}\sum_{D_j(\cdot)\in\mathbf{D}}P_{l_k}Q_j\widetilde{V}_{t+1}(i+\hat{x}-\hat{z},m^*)\right), \tag{34}$$

where $\widetilde{V}_{t+1}$ is an approximation of V_{t+1} in (20) given by

$$\widetilde{V}_{t+1}(i,m^*)\triangleq i\times\min_{D_j(\cdot)\in\mathbf{D}}D_j^{-1}(i) \tag{35}$$

for all i. That is, the value of inventory left at the end of period t is assumed to equal the revenue generated by selling that inventory at the lowest possible price.

We tested this heuristic on the fly using simulations. That is, problem (34) is solved, and the corresponding decision is implemented, only in states visited in a discrete-event simulation that samples price-demand functions and plant reliability values from their respective distributions, starting with zero initial inventory. We averaged the profits obtained in 10,000 such independent simulations to estimate the total expected profit generated by the one-step lookahead heuristic over periods $1,2,\ldots,T$. This estimate was only about 93% of the optimal value (recall in contrast from above that SFP values are on average 99% of the optimal). In addition, we ran 10,000 such simulations using a policy prescribed by SFP. A paired t-test showed that the difference between profits generated by the one-step lookahead heuristic and by SFP was statistically significant.

6 Conclusions and directions for future research

The main contribution of this paper is in showing rigorously how SFP, a game theoretic learning paradigm that has recently been successful in approximately solving other discrete optimization problems, can be applied to multi-action stochastic DPs. This is a class of stochastic sequential decision problems where the decision in each system state is an n-dimensional vector. These problems suffer from the curse of action-space dimensionality, because the effort needed for solving the Bellman's equation in each system state is exponential in n.

Our approach begins by viewing multi-action stochastic DPs as games of common interests among n players, each corresponding to one component of the decision vectors. Thus player i's pure strategy determines the ith component of the n-dimensional decision vector in each state. In each iteration of our simplified implementation of SFP, each player solves a best response problem where the pure strategies of the other players are fixed. Consequently, each of these best response problems is a DP with a one-dimensional action-space, and hence can be solved efficiently. This implies that whereas the computational effort required for exact solution of the original multi-action stochastic DP is exponential in n, the computational work done in a fixed number of SFP iterations is linear in n.

SFP's efficiency was demonstrated through numerical experiments on a case study from manufacturing. This manufacturing problem was a constrained multi-action stochastic DP. Specifically, (some of) the components of the multi-dimensional decision vectors were linked by coupling constraints and hence the DP did not have the Cartesian product structure that is essential for implementing SFP. We overcame this hurdle by developing a variable transformation that converted the constrained multi-action stochastic DP into an unconstrained one. SFP was then able to find near-optimal solutions hundreds of times faster than backward recursion. SFP also found significantly better solutions as compared to a one-step lookahead heuristic.

While we have laid a basic foundation for applying SFP to multi-action stochastic DPs, several open questions remain unanswered. These will provide directions for future research as we outline next.

6.1 Future research directions

Corollary 4.3 only guarantees convergence to the set of locally optimal policies. It is desirable to be able to strengthen this result and say that SFP converges to the set of optimal policies for multi-action stochastic DPs. However, such a strengthening does not appear possible at least with the standard version of SFP that was used in this paper. This limitation stems from the well-known fact that SFP may not in general converge to optimal Nash equilibria of games of common interests [16]. A more practically relevant question is whether the objective value of the best solution sampled by SFP converges to the optimal objective value. Unfortunately, even this weaker result does not hold for games of common interests. It is possible, however, to achieve this weaker but practically relevant form of convergence by implementing a "noisy" version of SFP where, in iteration k, each player randomly samples a policy with probability ϵ_k and samples from the empirical frequency of best responses as in Algorithm 4.1 with probability $1-\epsilon_k$. Then, if $\epsilon_k \to 0$ at a rate such that every pure joint strategy is sampled infinitely often with probability one as $k \to \infty$, it is easy to see that the objective value of the best policy sampled will converge to the optimal objective value. There has been a growing interest in such variants of SFP [8, 11, 20, 21, 30]. We leave their convergence analyses and computational comparisons in the specific context of multi-action stochastic DPs to future research.

To the best of our knowledge, all implementations of SFP, including the one in Section 5 in this paper, have used samples of size one, that is, $N_k = 1$. While this is a significant deviation from theory, it has proven adequate in efficiently finding good quality solutions in numerical experiments on static as well as dynamic optimization problems [8, 12, 17]. Using larger values of N_k in an SFP implementation leads to a better approximation of the expected value calculation in FP and also to a better exploration of the policy space, but it makes the best response computation more demanding. Compare, for instance, the simple best response DP (13) for $N_k = 1$ with the somewhat more clumsy best response DP (38) for $N_k > 1$. We did perform some exploratory numerical experiments with $N_k = 2, 3, 4, 5$ but found that $N_k = 1$ in fact achieved slightly better objective values. In the future, it would be interesting to numerically investigate this tradeoff using several values of N_k for various multi-action stochastic DPs.

As demonstrated in our manufacturing case study, onto transformations that map infeasible solutions into feasible ones significantly expand the applicability of our SFP algorithm to constrained multi-action stochastic DPs. For example, a simple transformation that converts the constrained multi-action stochastic DP formulation of a class of dynamic resource allocation problems into an unconstrained multi-action stochastic DP is described in Appendix B. While it is essentially always possible as stated in Section 2 to convert a constrained problem into an unconstrained one using a trivial onto transformation, we believe that effective transformations will need to be problem specific and may require some trial-and-error. Intuitively, "nondegenerate" transformations where componentwise variations in the constrained problem produce comparable componentwise variations in the unconstrained problem, are likely to work well. When the linking constraints define a full-dimensional, well-rounded (the ratio of the radii of a circumscribed and an inscribed ball is not too large) convex set in a Euclidean space and the feasible decision vectors in the constrained multi-action stochastic DP are given by the intersection of this convex set with the integer grid, one possibility could be to use proportional projection and rounding. Its rough schematic is shown in Figure 3. Future research will investigate the viability of such transformations and their impact on the quality of solutions delivered by SFP for different classes of constrained multi-action stochastic DPs.

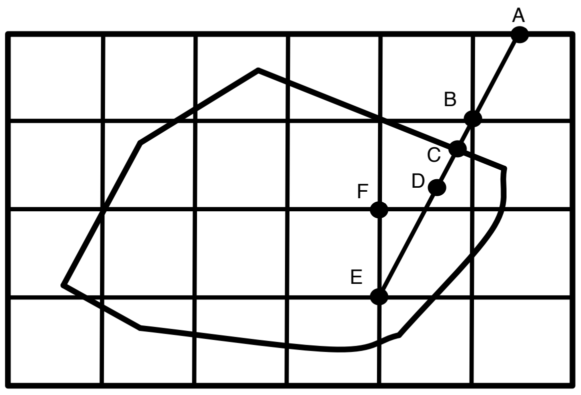

Figure 3: A schematic of projection and rounding when the feasible decision vectors are given by integer grid points inside a convex set. The bounding box grid represents the Cartesian product $A(s_t)$. E is an arbitrary but fixed feasible point known to the decision maker. The infeasible grid point B is first projected to point D such that the length ratios (BE/AE) and (DE/CE) are equal. Then point D is rounded to the feasible point F.

Acknowledgments

This research was funded in part by the National Science Foundation under grants CCF-0830380 and CCF-0830092, and by the General Motors Collaborative Research Laboratory at the University of Michigan. Archis Ghate appreciates summer support from the University of Washington.

A Formulation of a dynamic resource allocation problem as an unconstrained multi-action stochastic DP

We first provide a detailed description of a typical dynamic resource allocation problem. An initial amount $M > 0$ of one resource is available at the beginning of the first period. A random amount $b_t \in \{0, 1, 2 \ldots, B\}$ of this resource arrives at the beginning of period t, for $t = 2, \ldots, T$. The probability mass function of b_t is $q_t(\cdot)$. State s_t denotes the amount of resource available at the beginning of period t, and hence $s_1 = M$. Thus, for $t = 2, 3, \ldots, T$, the state space S_t equals

$\{0, 1, 2, \ldots, M + (t-1)B\}$. The cost of storing an amount s_t of the resource at the beginning of period t, for $t = 2, 3, \ldots, T+1$, is given by $h_t(s_t)$, where $h_t(\cdot)$ are the holding cost functions. The decision maker observes the amount of resource available and chooses positive integer levels $a_1, a_2, \ldots, a_n$ at which to perform n heterogeneous activities. Let $c_{1t} > 0, c_{2t} > 0, \ldots, c_{nt} > 0$ denote the amounts of the resource consumed in period t by each unit level of activities $1, 2, \ldots, n$, respectively. Let $A_{it}(s_t) = \{0, 1, 2, \ldots, \lfloor s_t/c_{it} \rfloor\}$, for $i = 1, 2, \ldots, n$. Here $\lfloor s_t/c_{it} \rfloor$ is the level at which activity i could be performed if all s_t units of the resource were allocated to this activity. Let $A_t(s_t) = A_{1t}(s_t) \times A_{2t}(s_t) \times \ldots \times A_{nt}(s_t)$. The set of feasible activity levels is then defined by

$$F_t(s_t) = \{(a_1, a_2, \ldots, a_n) \in A(s_t) : c_{1t}a_1 + c_{2t}a_2 + \ldots + c_{nt}a_n \le s_t\}.$$

The reward obtained on performing the n activities at levels $a = (a_1, \ldots, a_n) \in F_t(s_t)$ in period t is given by $\gamma_t(s_t, a)$, where $\gamma_t(\cdot, \cdot)$ is the reward function. After performing these activities, the state transitions into $s_{t+1} = s_t - (c_{1t}a_1 + c_{2t}a_2 + \ldots + c_{nt}a_n) + b_{t+1}$ with probability $q_{t+1}(b_{t+1})$. Thus the expected net reward accrued in period t is given by

$$r_t(s_t, a) = \gamma_t(s_t, a) - \sum_{b_{t+1}=0}^{B} q_{t+1}(b_{t+1}) h_{t+1}(s_t - (c_{1t}a_1 + c_{2t}a_2 + \ldots + c_{nt}a_n) + b_{t+1}).$$

The decision maker's goal is to choose a resource allocation policy, that is, to choose activity levels in each possible state in each period, so as to maximize the total expected net reward accrued in periods 1 through T. This problem is clearly a constrained multi-action stochastic DP.

This constrained problem can be converted into an unconstrained multi-action stochastic DP as follows. Instead of choosing decision vectors $a \in F_t(s_t)$, the decision maker chooses decision vectors $x \in A_t(s_t)$. We define transformations $\Omega_{t,s_t} : A_t(s_t) \to F_t(s_t)$ such that $\Omega_{t,s_t}(x) = y$, where

$$y = \left(\left\lfloor \frac{s_t x_1}{\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t)} \right\rfloor, \left\lfloor \frac{s_t x_2}{\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t)} \right\rfloor, \ldots, \left\lfloor \frac{s_t x_n}{\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t)} \right\rfloor \right).$$

Notice that if $x \in F_t(s_t)$, then $c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n \le s_t$ and hence $\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t) = s_t$. As a result, for any $i = 1, 2, \ldots, n$, we have $\frac{s_t x_i}{\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t)} = \frac{s_t x_i}{s_t} = x_i$.

On the other hand, if $x \in A_t(s_t) \setminus F_t(s_t)$, then $\max(c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n, s_t) = c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n$. Thus,

$$\begin{aligned} c_{1t}y_1 + \ldots + c_{nt}y_n &= c_{1t}\left\lfloor \frac{s_t x_1}{c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n} \right\rfloor + \ldots + c_{nt}\left\lfloor \frac{s_t x_n}{c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n} \right\rfloor \\ &\leq \frac{s_t c_{1t} x_1}{c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n} + \ldots + \frac{s_t c_{nt} x_n}{c_{1t}x_1 + c_{2t}x_2 + \ldots + c_{nt}x_n} = s_t. \end{aligned}$$

In summary, the transformations Ω_{t,s_t} do not alter feasible decision vectors but convert infeasible decision vectors into feasible ones. The former implies that these transformations are onto. If the decision maker chooses vector $x \in A_t(s_t)$, the system transitions into $s_{t+1} = s_t - (c_{1t}[\Omega_{t,s_t}(x)]_1 + \ldots + c_{nt}[\Omega_{t,s_t}(x)]_n) + b_{t+1}$ with probability $q_{t+1}(b_{t+1})$. We also extend the domain of the reward function $\gamma_t(\cdot,\cdot)$ from $S_t \times F_t(s_t)$ to $S_t \times A_t(s_t)$ by defining $\gamma_t(s_t, x) = \gamma_t(s_t, \Omega_{t,s_t}(x))$ for all $x \in A_t(s_t)$ and for all $s_t \in S_t$. Owing to the onto property of our transformations Ω_{t,s_t}, it is easy to see as in Lemma 3.1 that if an allocation policy is optimal to the unconstrained problem then the corresponding transformed policy is optimal to the constrained problem. Consequently, it suffices to solve the unconstrained problem to which the SFP algorithm described in this paper applies in a straightforward manner.

B Best response DP when the sample size $N_k > 1$

Consider the best response problem (11) for player i in Algorithm 4.1. For each aggregate pure joint strategy $\xi_{-i} \in \Psi_{-i}$ of all players other than player i, let $\kappa^k_{-i}(\xi_{-i}) = \{j : \psi^{jk}_{-i} = \xi_{-i}\}$. That is, $\kappa^k_{-i}(\xi_{-i})$ is the set of samples (out of the N_k samples drawn in iteration k) in which players other than player i sample ξ_{-i}. Interpret $\hat{f}^k_{-i}(\xi_{-i}) = \frac{|\kappa^k_{-i}(\xi_{-i})|}{N_k}$ as the probability with which players other than player i play the pure joint strategy ξ_{-i}. Player i's best response problem (11) can then be rewritten as

$$\psi^k_i = \underset{\psi_i \in \Psi_i}{\text{argmax}} \sum_{\xi_{-i} \in \Psi_{-i}} \frac{|\kappa^k_{-i}(\xi_{-i})|}{N_k} U(\psi_i, \xi_{-i}) = \underset{\psi_i \in \Psi_i}{\text{argmax}} \sum_{\xi_{-i} \in \Psi_{-i}} \hat{f}^k_{-i}(\xi_{-i}) U(\psi_i, \xi_{-i}) \tag{36}$$

$$= \underset{\psi_i \in \Psi_i}{\text{argmax}}\, V_{\Theta(\psi_i, \hat{f}^k_{-i})}(s_1). \tag{37}$$

Also let $\lambda_{-it}(s_t, a_{-i}; \hat{f}^k_{-i})$ be the probability that aggregate action $a_{-i} \in A_{-i}(s_t)$ is chosen in state $s_t \in S_t$ in period t by the aggregate player who employs mixed strategy $\hat{f}^k_{-i}$. That is,

$$\lambda_{-it}(s_t, a_{-i}; \hat{f}^k_{-i}) = \sum_{\{\xi_{-i} \in \Psi_{-i} : \xi_{-it}(s_t) = a_{-i}\}} \hat{f}^k_{-i}(\xi_{-i}).$$

Then problem (36) can be interpreted as follows. At the beginning of each period $t = 1, 2, \ldots, T$, player i observes the state $s_t \in S_t$ of the system and chooses a decision $a_i \in A_i(s_t)$. The system then transitions into a state $s_{t+1} \in S_{t+1}$ with probability $q^k_{it}(s_{t+1}; s_t, a_i)$ that is given by

$$q^k_{it}(s_{t+1}; s_t, a_i) = \sum_{a_{-i} \in A_{-i}(s_t)} p_t(s_{t+1}; s_t, (a_i, a_{-i}))\lambda_{-it}(s_t, a_{-i}; f^k_{-i}),$$

and player i receives an expected reward that equals $\sum_{a_{-i} \in A_{-i}(s_t)} \lambda_{-it}(s_t, a_{-i}; f^k_{-i}) r_t(s_t, (a_i, a_{-i}))$. Player i's goal is to maximize the total T-period expected reward it accrues starting in the initial state s_1. Problem (36) can therefore be formulated as a standard finite-horizon stochastic DP and can be solved through the following Bellman's equations for all $s_t \in S_t$ in the order $t = T, T-1, \ldots, 1$:

$$V^k_{it}(s_t) = \max_{a_i \in A_i(s_t)} \Bigg\{ \sum_{a_{-i} \in A_{-i}(s_t)} \lambda_{-it}(s_t, a_{-i}; f^k_{-i}) r_t(s_t, (a_i, a_{-i})) + \sum_{s_{t+1} \in S_{t+1}} q^k_{it}(s_{t+1}; s_t, a_i) V^k_{it+1}(s_{t+1}) \Bigg\}, \quad (38)$$

where $V^k_{iT+1}(s_{T+1}) = 0$ for all $s_{T+1} \in S_{T+1}$. Actions a_i that achieve the above maxima define a best response policy ψ^k_i for player i, which can then be employed to perform a belief update.

References

[1] D P Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, Nashua, NH, third edition, 2007.

[2] R I Brafman and M Tennenholtz. Learning to coordinate efficiently: A model-based approach. *Journal of Artificial Intelligence Research*, 19:11–23, 2003.

[3] G W Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. Wiley, 1951.

[4] E Campos-Nanez, A Garcia, and C Li. A game theoretic approach to efficient power management in sensor networks. *Operations Research*, 56(3):552–561, 2008.

[5] H S Chang, M C Fu, J Hu, and S I Marcus. *Sampling-based algorithms for Markov decision processes*. Springer-Verlag, London, UK, 2007.

[6] X D Chen and D Simchi-Levi. Coordinating inventory control and pricing strategies with random demand and fixed ordering costs: the finite horizon case. *Operations Research*, 52(6):887–896, 2004.

[7] S F Cheng, M A Epelman, and R L Smith. CoSIGN: A parallel algorithm for coordinated traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):551–564, 2006.

[8] M A Epelman, A Ghate, and R L Smith. Sampled Fictitious Play for approximate dynamic programming. *Computers and Operations Research*, 38:1705–1718, 2011.

[9] A A Federgruen and A Heching. Combined pricing and inventory control under uncertainty. *Operations Research*, 47(3):454–475, 1999.

[10] D Fudenberg and J Tirole. *Game Theory*. MIT Press, 1991.

[11] A Garcia, S D Patek, and K Sinha. A decentralized approach to discrete optimization via simulation: application to network flows. *Operations Research*, 55(4):717–732, 2007.

[12] A Garcia, D Reaume, and R L Smith. Fictitious play for finding system optimal routings in dynamic traffic networks. *Transportation Research Part B*, 34:147–156, 2000.

[13] G Infanger. Dynamic asset allocation strategies using a stochastic dynamic programming approach. In S A Zenios and W T Ziemba, editors, *Handbook of Asset and Liability Management*, volume 1, pages 200–251. Elsevier, 2006.

[14] S Kirkpatrick and C D Gelatt. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[15] L Y Koo, Y Chen, A Adhitya, R Srinivasan, and I A Karimi. Evaluating refinery supply chain policies and investment decisions through simulation-optimization. In *Proceedings of the Winter Simulation Conference*, 2006.

[16] T J Lambert, M A Epelman, and R L Smith. A fictitious play approach to large-scale optimization. *Operations Research*, 53(3):477–489, 2005.

[17] T J Lambert and H Wang. Fictitious play approach to a mobile unit situation awareness problem. Technical report, University of Michigan, Ann Arbor, Michigan, USA, 2003.

[18] C Maglaras. Revenue management for a multiclass single-server queue via a fluid model analysis. *Operations Research*, 54(5):914–932, 2006.

[19] C Maglaras and J Meissner. Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing and Service Operations Management*, 8(2):136–148, 2006.

[20] J R Marden, G Arslan, and J S Shamma. Joint strategy fictitious play with inertia for potential games. *IEEE Transactions on Automatic Control*, 54(2):208–220, 2009.

[21] J R Marden, H P Young, G Arslan, and J S Shamma. Payoff-based dynamics for multiplayer weakly acyclic games. *SIAM Journal of Control and Optimization*, 48(1):373–396, 2009.

[22] D Monderer and L Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68:258–265, 1996.

[23] W Powell. *Approximate Dynamic Programming.* John Wiley and Sons, New Jersy, 2007.

[24] M Puterman. *Markov Decision Processes.* John Wiley and Sons, New Jersy, 1994.

[25] J Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.

[26] S M Ross. *Introduction to stochastic dynamic programming.* Academic Press, New York, 1983.

[27] E Sisikoglu. *Distributed algorithms based on fictitious play for near optimal sequential decision making.* PhD thesis, University of Michigan, Ann Arbor, Michigan, USA, 2009.

[28] R S Sutton and A G Barto. *Reinforcement learning: an introduction.* MIT Press, Cambridge, Massachusetts, USA, 1998.

[29] K T Talluri and G van Ryzin. *The Theory and Practice of Revenue Management*. Springer, New York, 2005.

[30] H P Young. Evolution of conventions. *Econometrica*, 61(1):57–84, 1993.