Stochastic Optimization

Lauren A. Hannah

April 4, 2014

1 Introduction

Stochastic optimization refers to a collection of methods for minimizing or maximizing an objective function when randomness is present. Over the last few decades these methods have become essential tools for science, engineering, business, computer science, and statistics. Specific applications are varied, but include: running simulations to refine the placement of acoustic sensors on a beam, deciding when to release water from a reservoir for hydroelectric power generation, and optimizing the parameters of a statistical model for a given data set. Randomness usually enters the problem in two ways: through the cost function or the constraint set. Although stochastic optimization refers to any optimization method that employs randomness within some communities, we only consider settings where the objective function or constraints are random.

Like deterministic optimization, there is no single solution method that works well for all problems. Structural assumptions, such as limits on the size of the decision and outcome spaces, or convexity, are needed to make problems tractable. Since solution methods are tied to problem structure, we divide this chapter according to problem type and describe the associated solution methods.

The most prominent division is between solution methods for problems with a single time period (single stage problems) and those with multiple time periods (multistage problems). Single stage problems try to find a single, optimal decision, such as the best set of parameters for a statistical model given data. Multistage problems try to find an optimal sequence of decisions, such as scheduling water releases from hydroelectric plants over a two year period. Single stage problems are usually solved with modified deterministic optimization methods. However, the dependence of future decisions on random outcomes makes direct modification of deterministic methods difficult in multistage problems. Multistage methods are more reliant on statistical approximation and strong assumptions about problem structure, such as finite decision and outcome spaces, or a compact Markovian representation of the decision process. Section 2 describes solution methods for single stage stochastic optimization problems and Section 3 give methods for sequential problems. Conclusions and areas for advancement are given in Section 4.

2 Single Stage Stochastic Optimization

Single stage stochastic optimization is the study of optimization problems with a random objective function or constraints where a decision is implemented with no subsequent recourse. One example would be parameter selection for a statistical model: observations are drawn from an unknown distribution, giving a random loss for each observation. We wish to select model parameters to minimize the expected loss using data.

To generalize the problem, we begin by introducing some formal concepts and notation. Let $\mathcal{X}$ be the domain of all feasible decisions and x a specific decision. We would like to search over $\mathcal{X}$ to find a decision that minimizes a cost function, F. Let ξ denote random information that is available only after the decision is made. Unless otherwise noted, we will limit our discussion to random cost functions, denoted $F(x, \xi)$ (random constraint sets can be written as random cost functions). Since we cannot directly optimize $F(x, \xi)$, we instead minimize the expected value, $\mathbb{E}[F(x, \xi)]$. The general single stage stochastic optimization problem becomes

$$\zeta^* = \min_{x \in \mathcal{X}} \left\{ f(x) = \mathbb{E}[F(x, \xi)] \right\}. \tag{1}$$

Define the set of optima as $\mathcal{S}^* = \{x \in \mathcal{X} \, : \, f(x) = \zeta^*\}$. For all single stage problems, we assume that the decision space $\mathcal{X}$ is convex and the objective function $F(x, \xi)$ is convex in x for any realization ξ. Problems that do not meet these assumptions are usually solved through more specialized stochastic optimization methods like the stochastic ruler (Alrefaei & Andradottir 2001), nested partitions (Shi & Olafsson 2000), branch and bound (Norkin et al. 1998), or tabu search (Battiti & Tecchiolli 1996, Glover & Laguna 1999). In this section, we begin by outlining a set of problem considerations and then review four broad solution methods problem (1): sample average approximation, stochastic approximation, response surfaces, and metamodels.

2.1 Problem Considerations

Stochastic optimization has been studied in a broad set of communities that each developed methods to solve problems that were important to their own discipline. As such, there is no consensus about how data are generated or processed. Each community has a set of tacit assumptions, such as observational data with exogenous noise in statistical inference problems, or generative data with endogenous noise in engineering design problems. The types of assumptions are rarely stated, but we find it helpful to discuss them and how they interact with solution methods.

Noise generation: exogenous vs. endogenous. The random outcome ξ is exogenous if the decision x does not affect the distribution of ξ. Likewise, noise is endogenous if the choice of x may affect the distribution of ξ. An example of exogenous noise would be returns for a portfolio of stocks: portfolio choice will not affect stock market returns. Endogenous noise can arise in problems like acoustic sensor placement on a steel beam (Sadegh & Spall 1998): placement affects the recorded distortions.

Data generation: constructive vs. black box. An observation of $F(x, \xi)$ is constructive if, for a given ξ, $F(x, \xi)$ can be evaluated for any x. This is common when F is a parametric model, like a log likelihood of parameters x given data realization ξ. Conversely, $F(x, \xi)$ is generated by a black box if $F(x, \xi)$ can only be generated for a specific x. Constructive data generation is only possible if the noise is exogenous.

Data treatment: batch vs. online. Solution methods have batch data evaluation if all of the observations of $(x_i, F(x_i, \xi))_{i=1}^n$ are used at once to generate a decision. An example would be maximum likelihood estimation for a parametric statistical model. Solution methods have online data evaluation if the decision is updated in a sequential manner using one observation at a time.

Data type: observational vs. generative. Data are observational if $(x_i, F(x_i, \xi))_{i=1}^n$ are generated *a priori* and the decision maker cannot generate any more. An example would be using n observations to fit a parametric statistical model. Data are generative if the decision maker can obtain samples for a new decision value, $(x_i, F(x_i, \xi_i))$, as needed. This is common in experimental settings where new observations can be generated by changing experimental parameters and rerunning the experiment. A poorly studied but common middle ground is partially observational data, where a decision maker can obtain new samples by controlling a subset of the parameters. An example would be temperature data in a building that is controlled by a thermostat set point: the set point can be changed by the decision maker, but other variables that influence the optimal decision, like occupancy level, cannot be changed.

2.2 Sample Average Approximation

Sample average approximation (SAA) (Healy & Schruben 1991, Robinson 1996, Shapiro & Wardi 1996, Shapiro et al. 2002) is a two-part method that uses sampling and deterministic optimization to solve (1). The first step in SAA is sampling. While directly computing the expected cost function, $\mathbb{E}[F(x, \xi)]$, is not possible for most problems, it can be approximated through Monte Carlo sampling in some situations. Let $(\xi_i)_{i=1}^n$ be a set of independent, identically distributed realizations of ξ, and let $F(x, \xi_i)$ be the cost function realization for ξ_i. The expected cost function is approximated by the average of the realizations:

$$\mathbb{E}[F(x, \xi)] \approx \frac{1}{n} \sum_{i=1}^{n} F(x, \xi_i). \tag{2}$$

The second step in SAA is search. The right hand side of equation (2) is deterministic, so deterministic optimization methods can be used to solve the approximate problem:

$$\zeta_n^* = \min_{x \in \mathcal{X}} \left\{ f_n(x) = \frac{1}{n} \sum_{i=1}^{n} F(x, \xi_i) \right\}. \tag{3}$$

The set of approximate optima is $\mathcal{S}_n^* = \{x \in \mathcal{X} \, : \, f_n(x) = \zeta_n^*\}$.

Deterministic search is the main benefit of SAA. Many commercial software packages, including Matlab and R, offer implementation of basic deterministic optimization methods, while more specialized packages like CPLEX and Gurobi provide a wider array of deterministic methods. To guarantee convergence of the search method to a global optimum, it is assumed that $\mathcal{X}$ is convex and that $F(x, \xi_i)$ is convex in x for every realization of ξ. These assumptions can be lifted in exchange for a guarantee of only local optimality.

One of the main limitations is that SAA is only available for problems with exogenous noise and constructive data generation. It cannot accommodate any problems with black box generation. The data are processed in a batch manner. These rather strong assumptions allow the use of either observational or generative data.

Because of its inherent simplicity, SAA has been independently proposed in a variety of fields under a variety of names. A form of SAA was derived in the early statistics community as maximum likelihood estimation (Fisher 1922), although it was not generalized to other types of stochastic optimization problems. SAA-type methods were generalized in the operations research community under a variety of names: sample path optimization (Robinson 1996, Shapiro & Wardi 1996, Gürkan et al. 1999), stochastic counter methods (Rubinstein & Shapiro 1993), retrospective optimization (Healy & Schruben 1991), scenario optimization (Birge & Louveaux 1997), and sample average approximation (Shapiro et al. 2002, Kleywegt et al. 2002). Applications for SAA include: chance constrained optimization (Pagnoncelli et al. 2009), discrete stochastic optimization (Kleywegt et al. 2002), stochastic routing (Verweij et al. 2003), queuing models (Atlason et al. 2004), asset allocation (Blomvall & Shapiro 2006), and solving (Partially Observable) Markov Decision Processes ((PO)MDPs) (Ng & Jordan 2000, Perkins 2002).

2.2.1 Theoretical Properties of SAA Estimates

Like all stochastic optimization methods, SAA relies upon a collection of random variables to produce a statistical estimate. Most theoretical results follow directly from the the Law of Large Numbers and Central Limit Theorem due to the construction of SAA. Under mild regularity conditions, for any fixed x the limiting distribution of the SAA estimate is Gaussian with mean $f(x)$ and variance $\sigma^2(x) = \mathrm{Var}(F(x, \xi))/n < \infty$:

$$n^{\frac{1}{2}} \left[f_n(x) - f(x)\right] \xrightarrow{d} N\left(0, \sigma^2(x)\right).$$

The symbol $\xrightarrow{d}$ denotes "convergence in distribution." Under more stringent conditions, including Lipschitz continuity of $F(\cdot, \xi)$, convexity of $F(x, \xi)$ and $\mathcal{X}$, and $f(x)$ having a unique optimum $\mathcal{S}^* = \{x^*\}$, a similar result holds for the optimal values:

$$n^{\frac{1}{2}} \left[\zeta_n^* - \zeta^*\right] \xrightarrow{d} N\left(0, \sigma^2(x^*)\right). \tag{4}$$

The limiting Gaussian distribution in (4) can be used to determine the number of samples needed to generate an ϵ-optimal solution with at least probability $1-\alpha$. An ϵ-optimal solution is defined any solution $\tilde{x} \in \mathcal{X}$ such that $f(\tilde{x}) \leq \zeta^* + \epsilon$ for a specified $\epsilon > 0$. However, there is a tradeoff between the accuracy of f_n and the time needed to search f_n. In the general case, with efficient search algorithms the additional computational complexity is roughly on the order of n (Shapiro 2013). See Shapiro (1991), Rubinstein & Melamed (1998) and Shapiro et al. (2009) for a more complete discussion of SAA consistency and convergence rates.

2.3 Stochastic Approximation

Stochastic approximation is an iterative method that uses noisy observations to find the root of a function, g, *i.e.* the set of x where $g(x) = 0$. (Robbins & Monro 1951). When that function is the gradient of the expected cost function, $g(x) = \nabla_x f(x)$, stochastic approximation finds local optima of f for an unconstrained decision set using noisy subgradient observations, $g(x_i, \xi_i)$. In a generalization of stochastic gradient descent, a decision x_i is updated recursively by moving a distance a_i in the direction opposite the gradient:

$$x_{i+1} = x_i - a_i g(x_i, \xi_i). \tag{5}$$

A constraint set, $\mathcal{X}$, can be included through a projection operator $\Pi_{\mathcal{X}}$ that maps $x_i - a_i g(x_i, \xi_i)$ into the closest point of $\mathcal{X}$. The main considerations for (5) are selecting a step size sequence $(a_i)_{i=1}^{\infty}$ and obtaining a gradient $g(x_i, \xi_i)$ through direct observation or estimation.

Stochastic approximation has a less restrictive set of assumptions than SAA: data generation may be either constructive or black box, and noise generation may be either endogenous or exogenous. If the data have exogenous noise and constructive generation, then observational data may be used. If those conditions are not met, generative data are required. Finally, stochastic approximation uses online data processing. This has made stochastic approximation one of the most prominent methods for large scale statistical inference; see Bottou (2010), Hoffman et al. (2010), and Agarwal et al. (2012) for examples. For n observations, SAA requires n evaluations of the derivative of a log-likelihood function per iteration of the deterministic solver, which can become unreasonably expensive. In extremely large scale inference, it may even be problematic to store n observations in RAM. Stochastic approximation requires only n evaluations of the derivative of the log-likelihood function and allows data streaming from an external storage device.

2.3.1 Gradient Approximation

Gradients can be computed through either direct observation or statistical estimation. Direct computations are usually possible when the loss function has a closed form, as in case of parametric statistical inference. In contrast, the gradient must be estimated when the data are generated by a black box. Finite difference (Kiefer & Wolfowitz 1952) and related methods (Spall 1992) approximate the gradient by using the current decision, x_t, as a starting point and adding a perturbation, $e_j c_i$. Here e_j denotes a vector of all 0's with a 1 in the jth location and $(c_i)_{i=1}^{\infty}$ is a sequence of perturbation sizes that decreases to 0. The jth entry of the gradient is approximated by:

$$[g(x_i, \xi_i)]_j = \frac{F(x_i + c_i e_j, \xi_{i,1}) - F(x_i - c_i e_j, \xi_{i,2})}{2c_i}. \tag{6}$$

Estimates from (6) can be made more stable by using minibatches. Full finite difference computations may become infeasible when $F(x_i, \xi_i)$ is computationally expensive or p is large. Alternatives include one-sided differences, requiring $p + 1$ evaluations, and simultaneous perturbation analysis (Spall 1992), which generates an estimate by adding and subtracting a random perturbation vector to get an estimate with two observations.

2.3.2 Step Size Selection

The step size sequences $(a_i)_{i=1}^{\infty}$ and $(c_i)_{i=1}^{\infty}$ affect whether the stochastic approximation estimator converges to a locally optimal solution and the rate at which it converges. In general, convergence for stochastic approximation with a directly computed gradient is guaranteed if

$$\sum_{i=1}^{\infty} a_i = \infty, \qquad \sum_{i=1}^{\infty} a_i^2 < \infty, \qquad a_i > 0. \tag{7}$$

Generally, $(a_i)_{i=1}^{\infty}$ is chosen such that $a_i = Ci^{-\alpha}$ for $\alpha \in (\frac{1}{2}, 1]$ and a positive constant C. There is a tradeoff between the rate of convergence and robustness of the estimator: $\alpha = 1$ generally promises the best asymptotic rate in many circumstances, but is sensitive to misspecification of C or non-strong convexity of f. If second order information is available, a_i can be replaced by an approximation of the inverse of the Hessian of f (Bottou 2010).

When gradients are estimated, the perturbation sequence $(c_i)_{i=1}^{\infty}$ needs to be chosen in conjunction with $(a_i)_{i=1}^{\infty}$. Traditional constraints require that in addition to (7), the sequences satisfy (Kiefer & Wolfowitz 1952):

$$\sum_{i=1}^{\infty} \frac{a_i^2}{c_i^2} < \infty, \qquad \sum_{i=1}^{\infty} a_i c_i < \infty, \qquad c_i > 0.$$

However, slightly looser constraints do exist (Broadie et al. 2011). Parameters are commonly set as polynomial sequences with $a_i = Ci^{-1}$ and $c_i = Di^{-1/4}$ for positive constants C and D.

2.3.3 Theoretical Properties of Stochastic Approximation

Convergence theory for stochastic approximation generalizes convergence theory for gradient descent algorithms. Rates depend on the shape of f and whether the gradients are exact or estimated. Common shape assumptions are smoothness—whether f is Lipschitz and has Lipschitz first order derivatives—and strong convexity:

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq m||x - y||_2^2,$$

for all $x, y \in \mathcal{X}$, where $m > 0$ is a curvature constant. In all instances, we use n iterations and assume a few mild regularity conditions. If the gradients are not estimated, f is smooth and strongly convex, and $a_i = Ci^{-1}$, then the convergence rate is $\mathcal{O}(n^{-1})$:

$$\mathbb{E}\left[f(x_n) - f(x^*)\right] \leq Kn^{-1}$$

for a constant $K > 0$ if $mC > 2$. However, if $mC < 2$, then an arbitrarily poor rate $\mathcal{O}(n^{-mC/2})$ is obtained (Nemirovski et al. 2009, Bach & Moulines 2011). If the gradient is estimated, rates of $\mathcal{O}(n^{-1/2})$ can be obtained using $a_i = Ci^{-1}$ and $c_i = Di^{-1/4}$ (Broadie et al. 2011). When f is non-strongly convex or non-smooth, the best general convergence rate is $\mathcal{O}(n^{-1/2})$ (Bach & Moulines 2011).

Stochastic approximation can be made more robust to parameter misspecification through a combination of step sizes larger than $\mathcal{O}(n^{-1})$ and iterate averaging (Polyak & Juditsky

1992, Ruppert 1988, Nemirovski et al. 2009). The most common averaging scheme is Polayk-Ruppert averaging:

$$\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

where $x_1, \dots, x_n$ are generated through the usual stochastic approximation recursion. For a strongly convex, smooth objective function, a step size $a_i = Ci^{-\alpha}$ with $\alpha \in [1/2, 1]$ will yield rates of $\mathcal{O}(n^{-1})$ using iterate averaging (Bach & Moulines 2011). For convex, non-smooth or non-strongly convex objective functions, iterate averaging will give a rate of $\mathcal{O}(n^{-1/2})$ for $\alpha = 1/2$ (Nemirovski et al. 2009). In general, using $\alpha = 1/2$ and iterate averaging is a good choice because it is adaptive to strong convexity while being robust to non-smoothness and non-strong convexity.

2.4 Response Surfaces

Response surface methodology (RSM) is a collection of methods to methods that fit a surface to a set of decision—response pairs and then search the surface to find a new decision. Traditionally, RSM refers to online methods for decision making: a collection of decision—response pairs are sampled around a current decision, a "response surface" is fit locally to the observations, and then decision is updated by searching the local surface. Some authors have placed batch methods under the umbrella of RSM (Downing et al. 1985, Lim 2010), but we categorize these as global metamodels. RSM originated as methodology for optimizing conditions for chemical investigation in the 1950's (Box & Wilson 1951), and has since been widely applied in experimental design, process optimization, and simulation optimization. Recent surveys and books include Barton & Meckesheimer (2006), del Castillo (2007), Myers et al. (2009), and Barton (2013).

RSM proceeds in a manner similar to stochastic approximation. A starting point, x_0, is selected. Then configurations around this point are sampled and used to fit a first or second order polynomial (Box & Draper 1987). Expected values in a local area around the current decision x_i are estimated by:

$$\begin{aligned} \hat{y}(x_i) &= \beta_0 + \beta_{1:p}^T x_i, && \text{(linear)} \\ \hat{y}(x_i) &= \beta_0 + \beta_{1:p}^T x_i + x_i^T \mathbf{B} x_i. && \text{(quadratic)} \end{aligned}$$

Typically, a linear model is used in earlier iterations while a quadratic model is used in later ones. Sampling design near x_i is often based on experimental design literature for parametric models (Montgomery 2012) and model fit is evaluated with standard parametric statistical tools (Kleijnen & Sargent 2000). The decision is updated by moving from the current decision in the direction of steepest descent. For a locally linear model, this is

$$x_{i+1} = x_i - a_i \beta_{1:p}.$$

The step size a_i is often set as the quasi-Newton step size with the local surface as the objective function (Kleijnen 2008) or through a line search (Barton & Meckesheimer 2006). Once the estimated slopes get close to zero, a second order model is used to check optimality conditions through hypothesis testing.

Response surfaces have the same data generating requirements as stochastic approximation: data generation may be either constructive or black box, and noise generation may be either endogenous or exogenous. If the data have exogenous noise and constructive generation, then observational data may be used; otherwise, generative data are required.

2.5 Global Metamodel Optimization

Metamodels relate a set of inputs to an expected output through regression. Response surfaces are one form of metamodel, as are the Monte Carlo objective function approximations formed by SAA. In contrast to local response surfaces, global metamodels define a relationship that is valid across all feasible input values. A variety of regression methods have been used to fit global metamodels, including radial basis functions (Shin et al. 2002), neural networks (Anjum et al. 1997), Gaussian processes/stochastic kriging (Ankenman et al. 2010), and shape constrained nonparametric regression (Lim 2010, Hannah et al. 2013). Metamodels have been used for optimization for both batch and online problems.

In the batch setting, global metamodel optimization is done through a two-step process much like SAA. Before any modeling is done, n decision-response pairs, $(x_i, y_i)_{i=1}^n$ where $y_i = F(x_i, \xi_i)$, are collected. Then a metamodel, $f_n : \mathcal{X} \rightarrow \mathbb{R}$, is fit to $(x_i, y_i)_{i=1}^n$ through regression. Finally, the decision is generated by solving

$$\zeta_n^* = \min_{x \in \mathcal{X}} f_n(x).$$

Batch global metamodel optimization is primarily useful for inferring a decision from observational data generated by a black box with either endogenous or exogenous noise. Note that most regression methods, like neural networks, basis function regression, and Gaussian processes, are not shape-constrained: f_n is not guaranteed to be convex and must either be optimized locally or through global optimization methods. Recently, there has been some work on using shape-constrained regression to maintain metamodel convexity (Lim 2010, Hannah et al. 2013).

Global metamodel optimization has recently gained popularity for online optimization of functions that are costly to evaluate, such as hyperparameter tuning for Bayesian models (Scott et al. 2011, Snoek et al. 2012). The metamodels used are generative models, like Gaussian processes, which give a distribution over possible expected value functions. Design locations are then selected recursively to minimize an error metric, such as the expected minimal value given the new design location. After each design location is selected, the metamodel is updated. Selected design locations generally try to balance gaining knowledge about a region with high uncertainty against tuning decisions in an area that is known to perform well. Online global metamodels require generative data.

Along with potential search difficulties, global metamodel optimization is limited by the amount of data for the regression step. Nonparametric metamodels require that n grows exponentially with the dimension of $\mathcal{X}$. This issue can be bypassed by using models with strong assumptions, like parametric relationships.

3 Multistage Stochastic Optimization

Multistage stochastic optimization problems aim to find a sequence of decisions, $(x_t)_{t=0}^T$, that minimize an expected cost function. The subscript t denotes the time at which decision x_t is made. Usually decisions and random outcomes at time t affect the value of future decisions. An example would be making a move in a chess game. With a move, the player may capture one of her opponent's pieces, change her board position, and alter her possible future moves. She needs to account for these issues to select the move that maximizes her probability of winning. Mathematically, we can describe multistage stochastic optimization problems as an iterated expectation:

$$\zeta^* = \min_{x_0 \in \mathcal{X}_0} \mathbb{E}\left[\inf_{x_1 \in \mathcal{X}_1(x_0, \xi_1)} F_1(x_1, \xi_1) + \mathbb{E}\left[\cdots + \mathbb{E}\left[\inf_{x_T \in \mathcal{X}_T(x_{0:T-1}, \xi_{1:T})} \gamma^{T-1} F_T(x_T, \xi_T) \right]\right]\right]. \quad (8)$$

Here T is the number of time periods; $x_{0:t}$ is the collection of all decisions between 0 and t; ξ_t is a random outcome observable at time t; $\mathcal{X}_t(x_{0:t-1}, \xi_{1:t})$ is a decision set that depends on all decisions and random outcomes between times 0 and t; $F_t(x_t, \xi_t)$ is a cost function for time period t that depends on the decision and random outcome for period t; and γ is the discount rate. The time horizon T may be either finite or infinite.

Unlike the methods outlined in Section 2, there are no multistage solution methods that work well for all problems within a broad class, like convex problems or Markov decision processes. The decision sequence space is affected by the curse of dimensionality: the size of the space grows exponentially with T, the number of possible outcomes for ξ_t, and the size of the decision space each time period, $\mathcal{X}_t$. Most successful methods are tailored to problem subclasses with exploitable structure, including bandit problems and Markov decision processes (MDPs). In this section, we break describe solution methods according to problem classes: bandit problems, MDPs, and convex multistage problems.

3.1 Bandit Problems

Bandit problems are sequential decision problems where each time period a decision maker has a choice of playing one of K slot machines, or "one-armed bandits." Each machine has a payoff that can take values zero or one according to a Bernoulli distribution with an unknown, arm-dependent probability of success p_k. The goal of the decision maker is to maximize her expected revenue or to minimize regret after T time periods. The decision $x_{t-1} \in \{1, \dots, K\}$ is the selection of the arm to be played at time t, the random outcome ξ_t is a Bernoulli random variable with parameter $p_{x_{t-1}}$. The cost functions in equation (8) for maximizing expected rewards and minimizing regret are as follows:

$$F_t(x_{t-1}, \xi_t) = -\xi_t \qquad \text{(expected reward)}$$
$$F_t(x_{t-1}, \xi_t) = \mathbb{E}[\xi_t] - \max_{k=1,\dots,K} p_k. \qquad \text{(regret)}$$

Difficulty comes from the tradeoff between exploiting machines that have been observed to have high payouts and exploring new machines that have an uncertain payout probability. Solution methods can be divided by the size of K. When K is small each arm can be tried many times. When K is large the decision maker needs to borrow information from similar arms to estimate p_k, usually through regression. All bandit problems are online and assume generative data from a black box with endogenous noise.

Small K methods rely on computing a summary statistic for each arm, such as a mean value, upper confidence bounds (UCB) for p_k (Lai & Robbins 1985, Audibert et al. 2009), knowledge gradient (Frazier et al. 2008), or Gittins index (Gittins 1979). Once these have been computed, the arm with the highest value according to the statistic is played in the next round. These policies often result in asymptotic optimality.

When K is large, the arm space is often treated as a continuous space, $\mathcal{X}$. An arm mean function, $g(x)$, is assumed to have structure, often linearity, Lipschitz continuity, or a locally Lipschitz continuity. The ellipsoid method creates an ellipse that has high probability of containing the optimal solution, which works well with linear mean functions (Dani et al. 2008). UCBs can be generated for problems with Lipschitz or locally Lipschitz mean functions by using hierarchical models (Bubeck et al. 2011) or Gaussian processes (Srinivas et al. 2010).

Bandit problems can be extended to include "side information," which is an observable random variable that gives information about p_k. For instance, if p_k is an average click-through rate for an online advertisement, side information may be the advertiser, the user click history, zip code, etc. Bandit problems with side information are called "contextual bandits." Solution methods approximate the value of the arms as a function of the side information using regression methods like linear combinations of basis functions (Li et al. 2010), discretization of the state space (Rigollet & Zeevi 2010, Perchet & Rigollet 2013), random histograms (Yang & Zhu 2002), nearest neighbors (Yang & Zhu 2002) or adaptive partitioning (Slivkins 2011). While all bandit problems are broadly applicable to many e-commerce problems like advertising and page layout, contextual bandits are particularly useful models for online advertising when there is information about the user and/or the advertisement.

3.2 Markov Decision Processes

MDPs simplify (8) by assuming that the information required to make a decision at time t can be contained within a state, s_t. Naturally, this formulation includes all sequential decision processes the state is defined as the entirety of the decision and outcome history. However, MDPs are usually used to model problems where the state can take a small number of values that recur over time periods. For example, a state may be the position of a robot within a room where the goal is to navigate to a specific location while avoiding obstacles. Previous decisions, $x_{0:t}$, and random outcomes, $\xi_{1:t}$, are also summarized by the state. Due to the Markov nature of the problem, the cost function F_t is modified by removing the time index and including a dependence on the state to produce $F(x,s)$. The probability of transitioning from state s to state s' with decision x is $P_x(s,s')$. Traditionally, the time horizon is infinite. This allows optimality conditions to be expressed in terms of a value function $V(s)$, defined as the value of being in state s given that all subsequent decisions are optimal. The conditions are given by the Bellman optimality equation (Bellman 1961):

$$V(s) = \min_{x \in \mathcal{X}(s)} F(x,s) + \gamma \sum_{s'} P_x(s,s') V(s') = \min_{x \in \mathcal{X}(s)} F(x,s) + \gamma \mathbb{E}\left[V(s') \mid s, x\right]. \tag{9}$$

Since equation (9) is recursive, it is convenient to use value functions that are generated by a policy π, which maps each state to a decision:

$$V^{\pi}(s) = F(x, s) + \gamma \mathbb{E}\left[V^{\pi}(s') \mid s, x\right], \tag{10}$$

where x is chosen according to a policy π.

Solution methods for MDPs usually involve two steps: estimating value functions for a given policy π, and using then the estimated value functions to find a better policy π',

$$\pi'(s) = \arg\min_{x \in \mathcal{X}(s)} F(x, s) + \gamma \mathbb{E}\left[V^{\pi}(s') \mid s, x\right]. \tag{11}$$

Two of the most common solution methods are policy iteration and value iteration. Policy iteration selects a policy through (11), and then updates value function estimates in (10) using that policy. Value iteration combines the policy search and approximation steps by repeatedly solving:

$$\bar{V}_{i+1}(s) = \min_{x \in \mathcal{X}(s)} F(x, s) + \gamma \mathbb{E}\left[\bar{V}_i(s') \mid s, x\right], \tag{12}$$

where $\bar{V}_i$ is the estimated function after i iterations. See Sutton & Barto (1998) for a summary of these methods. Other solution techniques include rollout methods, which generate sample paths a few time periods into the future (Ng & Jordan 2000), Q-learning, which fits a regression model to state–decision pairs (Watkins & Dayan 1992), and dynamic programming, which is a computational method to solve finite horizon, discrete state, discrete decision problems (Sutton & Barto 1998). MDPs usually assume generative, black box data with endogenous noise. Most solution methods are online, although some methods like fitted-Q iteration (Murphy 2003, Antos et al. 2007) accommodate observational data in a batch setting.

MDP solution methods have many open research problems, including value function approximation, value attribution, and search over the decision space. Value function approximation is a form of regression, and it suffers the same problems as regression with regard to data sparsity in high dimensional state spaces. Proposed approximation methods have included lookup tables for small state spaces, linear combinations of a set of basis functions (Lagoudakis & Parr 2003), nonparametric functions (Engel et al. 2003), or other problem specific representations that include hierarchical structure or convexity (Powell 2011). Additionally, attributing value to actions when MDP has a single payout after a long string of decisions, such as winning a table tennis game, is notoriously difficult. Finally, all solution methods require a search over the decision space, as in (11) and (12). This is trivial when the decision space is finite and small, but can become difficult or impossible when the space is continuous and multidimensional.

3.3 Convex Multistage Problems

As in single stage optimization, convexity of the cost function and decision space allows decision makers to leverage deterministic convex optimization methods. Convexity occurs in many operations research problems like resource allocation, inventory management, and portfolio management. We define convex multistage problems with respect to equation (8):

$F_t(x_t, \xi_t)$ is convex in x_t for each ξ_t and $\mathcal{X}_t(x_{0:t-1}, \xi_{1:t})$ is convex for all $x_{0:t-1}$ and $\xi_{1:t}$. In almost all situations the time horizon T is finite and relatively small. Solution methods include stochastic programming and approximate dynamic programming (ADP).

Stochastic programming solves equation (8) by approximating it with a deterministic mathematical program, similar to SAA. One important but often unstated assumption is exogenous noise. It is described with a scenario tree. When n scenarios, $\{\omega_1, \dots, \omega_n\}$, are expressed as a tree, each scenario is a path from the current state to a leaf of the tree, $\{\xi_1(\omega_i), \dots, \xi_T(\omega_i)\}$. Scenario trees are often constructed through moment matching (Høyland & Wallace 2001) or Monte Carlo methods (Blomvall & Shapiro 2006), and each scenario can have nonuniform probability weights, $\{p_1(\omega_i), \dots, p_T(\omega_i)\}_{i=1}^n$. Decisions are also indexed by scenario, $x = \{x_0(\omega_i), \dots, x_T(\omega_i)\}_{i=1}^n$. The objective function is a linear combination of the costs for each of n scenarios:

$$f(x_{0:T}(\omega_1), \dots, x_{0:T}(\omega_n)) = \sum_{i=1}^{n} \sum_{t=1}^{T} p_t(\omega_i) F_t(x_t(\omega_i), \xi_t(\omega_i)). \tag{13}$$

As with SAA, this is a batch method using observational data and constructive observed costs. The tree structure is maintained through nonanticipativity constraints: if ω_i and ω_j share a node at time t, then $x_t(\omega_i) = x_t(\omega_j)$. Each decision also needs to satisfy the original constraints for a given scenario, $x_t(\omega_i) \in \mathcal{X}_t(x_{0:t-1}(\omega_i), \xi_{1:t}(\omega_i))$. A solution is found by minimizing (13) subject to the original and nonanticipativity constraints, often using dual decomposition methods (Ruszczyński 1997).

Stochastic programming methods have been successfully implemented for short and medium term scheduling problems, including reservoir management (Tejada-Guibert et al. 1995), unit commitment for electricity generation (Takriti et al. 2000), and asset management (Carino et al. 1994). Most computational difficulties come from large scenario trees, which occur when there are long time horizons, finer time scales, or outcome spaces that are not well described by a sparse tree.

ADP uses ideas from MDPs to solve (8), but it often has a finite time horizon and imposes shape constraints to allow search over continuous decision spaces. As in the MDP setting, the relevant historical decision and random outcome information is stored in a state variable, $(S_t)_{t=0}^T$, and the methods have the same data assumptions. ADP estimates a set of value functions to satisfy:

$$V_t(S_t) = \arg \min_{x_t \in \mathcal{X}_t(S_t)} \left\{ F_t(x_t, \xi_t) + \gamma \mathbb{E}\left[V(S_{t+1}) \mid S_t, x_t \right] \right\}. \tag{14}$$

As long as a convex approximation of $V_t(S_t)$ with respect to x_{t-1} is maintained, the convexity assumption allows efficient search over the decision space (Powell 2011). The value functions are often fit through methods similar to value iteration or policy iteration. While ADP can scale to much larger problems than stochastic programming (Anderson et al. 2011), approximating value functions is a nontrivial statistical challenge: state variables can be high dimensional, simulation outputs can be limited, and shape constrained inference is difficult in multiple dimensions (Hannah & Dunson 2013). Approximation methods have included basis functions linear projections (Tsitsiklis & Van Roy 2001, de Farias & Van Roy 2003, Keller et al. 2006), Benders cuts (Higle & Sen 1991, Zakeri et al. 2000), and nonparametric regression (George et al. 2008, Deisenroth et al. 2009, Taylor & Parr 2009). Consistency of ADP has not been shown in a general setting, although it has been demonstrated for certain cases (Nascimento & Powell 2009).

4 Future Challenges and Closing Remarks

Due to the swift progress of data storage and analysis capabilities, stochastic optimization has been an area of extremely active research. Summaries of progress within specific fields are given by Fu (2013), Hutchison & Spall (2013), and Powell (2014). However, many important areas lacking solutions include the following.

Efficient methods for black box, observational data. Observational data are currently handled through a global regression over the decision space, and problems with partially observational data receive little study. Methods that exploit convexity could drastically reduce data requirements and provide more reliable search.

Efficient methods for large multistage problems. While much attention has been given to value function approximation, discovering new problem subclasses with exploitable structure may prove more useful. Possible areas include the interplay between large decisions, such as three year operating goals for a hydroelectric plant, and small decisions, such as hourly water releases.

Solution evaluation methods. Asymptotic theory provides estimates on the quality of single stage solutions, but in the multistage case asymptotic bounds either do not exist or they only provide pessimistic bounds. Some work in the operations research literature uses information relaxations to provide dual bounds (Haugh & Kogan 2004, Brown et al. 2010), but these are computationally infeasible for general problems. Fast, reliable bounds would be useful for practitioners to assess decision quality before implementation.

Optimization in the presence of randomness is a fundamental problem in many fields. We fully expect that current methods for stochastic optimization will be refined over the next few decades while computational advances and new problems will lead to new opportunities for stochastic optimization research.

References

Agarwal, A., Negahban, S. & Wainwright, M. J. (2012), ‘Fast global convergence of gradient methods for high-dimensional statistical recovery’, *The Annals of Statistics* **40**(5), 2452–2482.

Alrefaei, M. H. & Andradottir, S. (2001), ‘A modification of the stochastic ruler method for discrete stochastic optimization’, *European Journal of Operational Research* **133**(1), 160–182.

Anderson, R. N., Boulanger, A., Powell, W. B. & Scott, W. (2011), 'Adaptive stochastic control for the smart grid', *Proceedings of the IEEE* **99**(6), 1098–1115.

Anjum, M. F., Tasadduq, I. & Al-Sultan, K. (1997), 'Response surface methodology: A neural network approach', *European Journal of Operational Research* **101**(1), 65–73.

Ankenman, B., Nelson, B. L. & Staum, J. (2010), 'Stochastic kriging for simulation metamodeling', *Operations Research* **58**(2), 371–382.

Antos, A., Munos, R. & Szepesvári, C. (2007), Fitted Q-iteration in continuous action-space MDPs, *in* J. Platt, D. Koller, Y. Singer & S. Roweis, eds, 'Advances in Neural Information Processing Systems 20', Vancouver, B.C., pp. 9–16.

Atlason, J., Epelman, M. A. & Henderson, S. G. (2004), 'Call center staffing with simulation and cutting plane methods', *Annals of Operations Research* **127**(1-4), 333–358.

Audibert, J.-Y., Munos, R. & Szepesvári, C. (2009), 'Exploration–exploitation tradeoff using variance estimates in multi-armed bandits', *Theoretical Computer Science* **410**(19), 1876–1902.

Bach, F. & Moulines, E. (2011), Non-asymptotic analysis of stochastic approximation algorithms for machine learning, *in* J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira & K. Weinberger, eds, 'Advances in Neural Information Processing Systems 24', Granada, Spain, pp. 451–459.

Barton, R. R. (2013), Response surface methodology, *in* S. I. Gass & M. Fu, eds, 'Encyclopedia of Operations Research and Management Science', third edn, Springer, New York.

Barton, R. R. & Meckesheimer, M. (2006), Metamodel-based simulation optimization, *in* S. G. Henderson & B. L. Nelson, eds, 'Simulation', Vol. 13 of *Handbooks in Operations Research and Management Science*, Elsevier, pp. 535–574.

Battiti, R. & Tecchiolli, G. (1996), 'The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization', *Annals of Operations Research* **63**(2), 151–188.

Bellman, R. (1961), *Adaptive Control Processes: A Guided Tour*, Vol. 4, Princeton University Press, Princeton, NJ.

Birge, J. R. & Louveaux, F. (1997), *Introduction to Stochastic Programming*, second edn, Springer Verlag, New York.

Blomvall, J. & Shapiro, A. (2006), 'Solving multistage asset investment problems by the sample average approximation method', *Mathematical Programming* **108**(2), 571–595.

Bottou, L. (2010), Large-scale machine learning with stochastic gradient descent, *in* Y. Lechevallier & G. Saporta, eds, 'Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT 2010)', Springer, Paris, France, pp. 177–187.
URL: *http://leon.bottou.org/papers/bottou-2010*

Box, G. E. P. & Draper, N. R. (1987), *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, Hoboken, NJ.

Box, G. E. P. & Wilson, K. B. (1951), 'On the experimental attainment of optimum conditions', *Journal of the Royal Statistical Society, Series B* **13**(1), 1–45.

Broadie, M., Cicek, D. & Zeevi, A. (2011), 'General bounds and finite-time improvement for the Kiefer-Wolfowitz stochastic approximation algorithm', *Operations Research* **59**(5), 1211–1224.

Brown, D., Smith, J. & Sun, P. (2010), 'Information relaxations and duality in stochastic dynamic programs', *Operations Research* **58**(4), 785–801.

Bubeck, S., Munos, R., Stoltz, G. & Szepesvari, C. (2011), 'X-armed bandits', *The Journal of Machine Learning Research* **12**, 1655–1695.

Carino, D. R., Kent, T., Myers, D. H., Stacy, C., Sylvanus, M., Turner, A. L., Watanabe, K. & Ziemba, W. T. (1994), 'The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming', *Interfaces* **24**(1), 29–49.

Dani, V., Hayes, T. P. & Kakade, S. M. (2008), Stochastic linear optimization under bandit feedback., *in* R. Servedio & T. Zhang, eds, 'Proceedings of the 21st Annual Conference on Learning Theory', Helsinki, Finland, pp. 355–366.

de Farias, D. P. & Van Roy, B. (2003), 'The linear programming approach to approximate dynamic programming', *Operations Research* **51**(6), 850–865.

Deisenroth, M. P., Rasmussen, C. E. & Peters, J. (2009), 'Gaussian process dynamic programming', *Neurocomputing* **72**(7), 1508–1524.

del Castillo, E. (2007), *Process Optimization: A Statistical Approach*, Springer, New York.

Downing, D. J., Gardner, R. H. & Hoffman, F. O. (1985), 'An examination of response-surface methodologies for uncertainty analysis in assessment models', *Technometrics* **27**(2), 151–163.

Engel, Y., Mannor, S. & Meir, R. (2003), Bayes meets Bellman: The Gaussian process approach to temporal difference learning, *in* T. Fawcett & N. Mishra, eds, 'Proceedings of the 20th International Conference on Machine Learning', Washington, D.C., pp. 154–161.

Fisher, R. A. (1922), 'On the mathematical foundations of theoretical statistics', *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **222**, 309–368.

Frazier, P. I., Powell, W. B. & Dayanik, S. (2008), 'A knowledge-gradient policy for sequential information collection', *SIAM Journal on Control and Optimization* **47**(5), 2410–2439.

Fu, M. C. (2013), Simulation optimization, *in* S. I. Gass & M. Fu, eds, 'Encyclopedia of Operations Research and Management Science', third edn, Springer, New York.

Gass, S. I. & Fu, M., eds (2013), *Encyclopedia of Operations Research and Management Science*, Springer, New York.

George, A., Powell, W. B. & Kualkarni, S. R. (2008), 'Value function approximation using multiple aggregation for multiattribute resource management.', *Journal of Machine Learning Research* **9**(10), 2079–2111.

Gittins, J. C. (1979), 'Bandit processes and dynamic allocation indices', *Journal of the Royal Statistical Society: Series B* pp. 148–177.

Glover, F. & Laguna, M. (1999), *Tabu Search*, Springer, New York.

Gürkan, G., Yonca Özge, A. & Robinson, S. M. (1999), 'Sample-path solution of stochastic variational inequalities', *Mathematical Programming* **84**(2), 313–333.

Hannah, L. A. & Dunson, D. B. (2013), 'Multivariate convex regression with adaptive partitioning', *Journal of Machine Learning Research* **14**, 3207–3240.

Hannah, L. A., Powell, W. B. & Dunson, D. B. (2013), 'Semi-convex regression for metamodeling-based optimization', *SIAM Journal on Optimization* . to appear.

Haugh, M. B. & Kogan, L. (2004), 'Pricing American options: a duality approach', *Operations Research* **52**(2), 258–270.

Healy, K. & Schruben, L. W. (1991), Retrospective simulation response optimization, *in* 'Proceedings of the 1991 Winter Simulation Conference, 1991', pp. 901–906.

Higle, J. & Sen, S. (1991), 'Stochastic decomposition: An algorithm for two-stage linear programs with recourse', *Mathematics of Operations Research* **16**(3), 650–669.

Hoffman, M., Bach, F. R. & Blei, D. M. (2010), Online learning for latent Dirichlet allocation, *in* J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel & A. Culotta, eds, 'Advances in Neural Information Processing Systems 23', Vancouver, B.C., pp. 856–864.

Høyland, K. & Wallace, S. (2001), 'Generating scenario trees for multistage decision problems', *Management Science* **47**(2), 295–307.

Hutchison, D. W. & Spall, J. C. (2013), Stochastic approximation, *in* S. I. Gass & M. Fu, eds, 'Encyclopedia of Operations Research and Management Science', third edn, Springer, New York.

Keller, P. W., Mannor, S. & Precup, D. (2006), Automatic basis function construction for approximate dynamic programming and reinforcement learning, *in* W. Cohen & A. Moore, eds, 'Proceedings of the 23rd International Conference on Machine Learning', Pittsburgh, PA, pp. 449–456.

Kiefer, J. & Wolfowitz, J. (1952), 'Stochastic estimation of the maximum of a regression function', *The Annals of Mathematical Statistics* **23**(3), 462–466.

Kleijnen, J. P. C. (2008), 'Response surface methodology for constrained simulation optimization: an overview', *Simulation Modelling Practice and Theory* **16**(1), 50–64.

Kleijnen, J. P. & Sargent, R. G. (2000), 'A methodology for fitting and validating metamodels in simulation', *European Journal of Operational Research* **120**(1), 14–29.

Kleywegt, A. J., Shapiro, A. & Homem-de Mello, T. (2002), 'The sample average approximation method for stochastic discrete optimization', *SIAM Journal on Optimization* **12**(2), 479–502.

Lagoudakis, M. G. & Parr, R. (2003), 'Least-squares policy iteration', *The Journal of Machine Learning Research* **4**, 1107–1149.

Lai, T. L. & Robbins, H. (1985), 'Asymptotically efficient adaptive allocation rules', *Advances in Applied Mathematics* **6**(1), 4–22.

Li, L., Chu, W., Langford, J. & Schapire, R. (2010), A contextual-bandit approach to personalized news article recommendation, *in* M. Rappa, P. Jones, J. Freire & S. Chakrabarti, eds, 'Proceedings of the 19th International Conference on the World Wide Web', Raleigh, NC, pp. 661–670.

Lim, E. (2010), Response surface computation via simulation in the presence of convexity constraints, *in* B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan & E. Yücesan, eds, 'Proceedings of the 2010 Winter Simulation Conference', Baltimore, MD, pp. 1246–1254.

Montgomery, D. C. (2012), *Design and Analysis of Experiments*, Eighth edn, Wiley, Hoboken, NJ.

Murphy, S. A. (2003), 'Optimal dynamic treatment regimes', *Journal of the Royal Statistical Society: Series B* **65**(2), 331–355.

Myers, R. H., Montgomery, D. C. & Anderson-Cook, C. M. (2009), *Response surface methodology: process and product optimization using designed experiments*, John Wiley & Sons, Hoboken, NJ.

Nascimento, J. & Powell, W. (2009), 'An optimal approximate dynamic programming algorithm for the lagged asset acquisition problem', *Mathematics of Operations Research* **34**(1), 210–237.

Nemirovski, A., Juditsky, A., Lan, G. & Shapiro, A. (2009), 'Robust stochastic approximation approach to stochastic programming', *SIAM Journal on Optimization* **19**(4), 1574–1609.

Ng, A. Y. & Jordan, M. (2000), PEGASUS: A policy search method for large MDPs and POMDPs, *in* K. B. Laskey, C. Boutilier & M. Goldszmidt, eds, 'Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence', Stanford, CA, pp. 406–415.

Norkin, V. I., Pflug, G. C. & Ruszczynski, A. (1998), 'A branch and bound method for stochastic global optimization', *Mathematical Programming* **83**(1-3), 425–450.

Pagnoncelli, B. K., Ahmed, S. & Shapiro, A. (2009), 'Sample average approximation method for chance constrained programming: theory and applications', *Journal of Optimization Theory and Applications* **142**(2), 399–416.

Perchet, V. & Rigollet, P. (2013), 'The multi-armed bandit problem with covariates', *The Annals of Statistics* **41**(2), 693–721.

Perkins, T. J. (2002), Reinforcement learning for POMDPs based on action values and stochastic optimization, *in* R. Dechter, M. Kearns & R. Sutton, eds, 'Proceedings of the 18th National Conference on Artificial Intelligence', Edmonton, Alberta, pp. 199–204.

Polyak, B. T. & Juditsky, A. B. (1992), 'Acceleration of stochastic approximation by averaging', *SIAM Journal on Control and Optimization* **30**(4), 838–855.

Powell, W. B. (2011), *Approximate Dynamic Programming: Solving the curses of dimensionality*, Second edn, John Wiley & Sons, Hoboken, NJ.

Powell, W. B. (2014), 'Energy and uncertainty: Models and algorithms for complex energy systems', *AI Magazine* p. to appear.

Rigollet, P. & Zeevi, A. (2010), Nonparametric bandits with covariates, *in* A. T. Kalai & M. Mohri, eds, 'Proceedings of the 23rd Annual Conference on Learning Theory', Haifa, Israel, pp. 54–66.

Robbins, H. & Monro, S. (1951), 'A stochastic approximation method', *The Annals of Mathematical Statistics* **22**(3), 400–407.

Robinson, S. M. (1996), 'Analysis of sample-path optimization', *Mathematics of Operations Research* **21**(3), 513–528.

Rubinstein, R. Y. & Melamed, B. (1998), *Modern Simulation and Modeling*, Wiley, Hoboken, NJ.

Rubinstein, R. Y. & Shapiro, A. (1993), *Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method*, John Wiley & Sons Inc, Hoboken, NJ.

Ruppert, D. (1988), Efficient estimations from a slowly convergent Robbins-Monro process, Technical report, Cornell University Operations Research and Industrial Engineering.

Ruszczyński, A. (1997), 'Decomposition methods in stochastic programming', *Mathematical Programming* **79**(1), 333–353.

Sadegh, P. & Spall, J. C. (1998), Optimal sensor configuration for complex systems, *in* J. H. Chow, ed., 'Proceedings of the 1998 American Control Conference', Philadelphia, PA, pp. 376–380.

Scott, W., Frazier, P. & Powell, W. (2011), 'The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression', *SIAM Journal on Optimization* **21**(3), 996–1026.

Shapiro, A. (1991), ‘Asymptotic analysis of stochastic programs’, *Annals of Operations Research* **30**(1), 169–186.

Shapiro, A. (2013), Sample average approximation, *in* S. I. Gass & M. Fu, eds, ‘Encyclopedia of Operations Research and Management Science’, third edn, Springer, New York.

Shapiro, A., Dentcheva, D. & Ruszczyński, A. (2009), *Lectures on Stochastic Programming: Modeling and Theory*, Society for Industrial Mathematics, Philadelphia.

Shapiro, A., Homem-de Mello, T. & Kim, J. (2002), ‘Conditioning of convex piecewise linear stochastic programs’, *Mathematical Programming* **94**(1), 1–19.

Shapiro, A. & Wardi, Y. (1996), ‘Convergence analysis of stochastic algorithms’, *Mathematics of Operations Research* **21**(3), 615–628.

Shi, L. & Olafsson, S. (2000), ‘Nested partitions method for stochastic optimization’, *Methodology in Computing and Applied Probability* **2**(3), 271–291.

Shin, M., Sargent, R. G. & Goel, A. L. (2002), Gaussian radial basis functions for simulation metamodeling, *in* J. L. Snowdon & J. M. Charnes, eds, ‘Proceedings of the 2002 Winter Simulation Conference’, San Diego, CA, pp. 483–488.

Slivkins, A. (2011), Contextual bandits with similarity information, *in* S. Kakade & U. von Luxborg, eds, ‘Proceedings of the 24th Annual Conference on Learning Theory’, Budapest, Hungary, pp. 679–701.

Snoek, J., Larochelle, H. & Adams, R. (2012), Practical Bayesian optimization of machine learning algorithms, *in* ‘Advances in Neural Information Processing Systems’, pp. 2960–2968.

Spall, J. C. (1992), ‘Approximation using a simultaneous perturbation gradient approximation’, *IEEE Transactions on Automatic Control* **37**(3), 332–341.

Srinivas, N., Krause, A., Kakade, S. M. & Seeger, M. (2010), Gaussian process optimization in the bandit setting: no regret and experimental design, *in* J. Fürnkranz & T. Joachims, eds, ‘Proceedings of the 27th International Conference on Machine Learning’, Haifa, Israel, pp. 1015–1022.

Sutton, R. S. & Barto, A. G. (1998), *Introduction to Reinforcement Learning*, MIT Press, Cambridge.

Takriti, S., Krasenbrink, B. & Wu, L. S.-Y. (2000), ‘Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem’, *Operations Research* **48**(2), 268–280.

Taylor, G. & Parr, R. (2009), Kernelized value function approximation for reinforcement learning, *in* L. Bottou & M. Littman, eds, ‘Proceedings of the 26th International Conference on Machine Learning’, Montreal, Quebec, pp. 1017–1024.

Tejada-Guibert, J. A., Johnson, S. A. & Stedinger, J. R. (1995), 'The value of hydrologic information in stochastic dynamic programming models of a multireservoir system', *Water Resources Research* **31**(10), 2571–2579.

Tsitsiklis, J. N. & Van Roy, B. (2001), 'Regression methods for pricing complex American-style options', *IEEE Transactions on Neural Networks* **12**(4), 694–703.

Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G. & Shapiro, A. (2003), 'The sample average approximation method applied to stochastic routing problems: a computational study', *Computational Optimization and Applications* **24**(2-3), 289–333.

Watkins, C. J. & Dayan, P. (1992), 'Q-learning', *Machine Learning* **8**(3-4), 279–292.

Yang, Y. & Zhu, D. (2002), 'Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates', *Annals of Statistics* **30**(1), 100–121.

Zakeri, G., Philpott, A. B. & Ryan, D. M. (2000), 'Inexact cuts in Benders decomposition', *SIAM Journal on Optimization* **10**(3), 643–657.