Simulation optimization: A review of algorithms and applications

Satyajith Amaran[1,2], Nikolaos V. Sahinidis[1], Bikram Sharda[3], and Scott J. Bury[4]

[1]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Now at Core R&D, The Dow Chemical Company, Freeport, TX 77541, USA
[3]Core R&D, The Dow Chemical Company, Freeport, TX 77541, USA
[4]Core R&D, The Dow Chemical Company, Midland, MI 48667, USA

Abstract

Simulation Optimization (SO) refers to the optimization of an objective function subject to constraints, both of which can be evaluated through a stochastic simulation. To address specific features of a particular simulation—discrete or continuous decisions, expensive or cheap simulations, single or multiple outputs, homogeneous or heterogeneous noise—various algorithms have been proposed in the literature. As one can imagine, there exist several competing algorithms for each of these classes of problems. This document emphasizes the difficulties in simulation optimization as compared to mathematical programming, makes reference to state-of-the-art algorithms in the field, examines and contrasts the different approaches used, reviews some of the diverse applications that have been tackled by these methods, and speculates on future directions in the field.

Keywords: Simulation optimization; Optimization via simulation; Derivative-free optimization

1 Introduction

Advances in modeling and availability of cheap computational power have enabled the science, engineering, and business research communities to make use of simulations to model phenomena and systems. It is only natural that there be a great interest in manipulating degrees of freedom in the simulations to optimize them.

The term Simulation Optimization (SO) is an umbrella term for techniques used to optimize stochastic simulations. Simulation optimization involves the search for those specific settings of the input parameters to a stochastic simulation such that a target objective, which is a function of the simulation output, is, without loss of generality, minimized.

As opposed to mathematical programming, SO does not assume that an algebraic description of the simulation is available—the simulation may be available as a black box that only allows the evaluation of the objective and constraints for a particular input. In fact, many SO algorithmic approaches solely depend on such input-output data from the simulation in their search for optimal input settings.

In addition, many large-scale and/or detailed simulations may be expensive to run, in terms of time, money, or resources. As a result, there is also a need to perform few simulations in this search for optimal parameters. Outputs from these stochastic simulations are not deterministic, and

usually follow some output distribution, which may or may not vary across the parametric space. This uncertainty or variability in output also adds to the challenge of optimization, as it becomes harder to discern the quality of the parametric input in the presence of this output noise. In addition, when an algebraic description of the simulation is not accessible, derivative information is usually unavailable, and the estimation of derivatives from the use of finite differences may not be suitable due to noisy outputs and the expensive nature of simulations.

The nature of the stochastic simulations under study will determine the specific technique chosen to optimize them. The simulations, which are often discrete-event simulations, may be partially accessible to us in algebraic form, or may be purely available as an input-output model (as a black box); they may have single or multiple outputs; they may have deterministic or stochastic output(s); they may involve discrete or continuous parameters; and they may or may not involve explicit, or even implicit/hidden constraints.

A very general simulation optimization problem can be represented by (P1).

$$
\begin{aligned}
\min \quad & \mathbb{E}_\omega[f(x, y, \omega)] \\
\text{s.t.} \quad & \mathbb{E}_\omega[g(x, y, \omega)] \leq 0 \\
& h(x, y) \leq 0 \\
& x_l \leq x \leq x_u \\
& x \in \mathbb{R}^n, y \in \mathbb{D}^m.
\end{aligned}
\tag{P1}
$$

The function f can be evaluated through simulation for a particular instance of the continuous inputs x, discrete inputs y, and a realization of the random variables in the simulation, the vector ω (which may or may not be a function of the inputs, x and y). Similarly, the constraints defined by the vector-valued function g are also evaluated with each simulation run. In this formulation, expected values for these stochastic functions are used. There may be other constraints (represented by h) that do not involve random variables, as well as bound constraints on the decision variables.

The relaxation of any of these conditions would constitute a problem that would fall under the purview of SO. Most algorithms focus on problems that either have solely discrete choices, or solely continuous decisions to make. Each constraint may be thought of as representing additional outputs from the simulation that need to be taken into consideration. In addition, there may be bound constraints imposed on decision variables, that may either be available or obtained from domain-specific

knowledge. Relatively few existing algorithms attempt to address both discrete and continuous choices simultaneously, although some broad classes of approaches naturally lend themselves to be applicable in either, and therefore both, settings. Further, the discrete variables may either be binary, integer-ordered, or categorical and lie in some discrete space $\mathbb{D}$.

As can be seen, the formulation P1 is extremely general, and therefore a wide variety of applications fall under the scope of simulation optimization. Various applications of simulation optimization in diverse research fields are tabulated in Section 2.

Another common assumption is that f is a real-valued function and g is a real vector-valued function, both of whose expected values may or may not be smooth or continuous functions. The most common objective in SO is to optimize the expected value of some performance metric, but other objective functions may be appropriate depending on the application. For instance, an objective that minimizes risk could be a possible alternative, in which case one would incorporate some sort of variance measure as well into the objective.

This paper is an updated version of Amaran et al. [4] and is meant to be a survey of available techniques as well as recent advances in simulation optimization. The remainder of the introduction section provides a literature survey of prior reviews, and elaborates on the relationship of simulation optimization to mathematical programming, derivative-free optimization, and machine learning. Section 2 provides a glimpse into the wide variety of applications of simulation optimization that have appeared in the literature. Section 3 focuses on various algorithms for discrete and continuous simulation optimization, provides basic pseudocode for major categories of algorithms, and provides comprehensive references for each type of algorithm. Section 4 provides a listing of available software for simulation optimization and Section 5 discusses means to compare their performance. Section 6 summarizes the progress of the field, and outlines some current and future topics for research.

1.1 Prior reviews of simulation optimization

Several review papers (cf. [135, 99, 174, 15, 59, 29, 7, 16, 191, 61, 60, 194, 62, 9, 88, 5, 152]), books and research monographs (cf. [186, 171, 115, 35]), and theses (cf. [12, 53, 45, 30, 58, 104]) have traced the development of simulation optimization.

Meketon [135] provides a classification of algorithmic approaches for optimization over simulations based on how much information or structure about

the underlying model is known. The paper surveys the progress of the field between 1975 and 1987, and focuses on continuous simulation optimization. Andradóttir [7] provides a tutorial on gradient-based procedures for continuous problems. Carson and Maria [29] and Azadivar [16] also give brief outlines of and pointers to prevailing simulation optimization algorithms.

Fu et al. [61] contains several position statements of eminent researchers and practitioners in the field of simulation, where the integration of simulation with optimization is discussed. The issues addressed include generality vs. specificity of an algorithm, the wider scope of problems that simulation optimization methodologies have the potential to address, and the need for integrating provably convergent algorithms proposed by the research community with metaheuristics often used by commercial simulation software packages.

Of the more recent surveys, Fu [59] provides an excellent tutorial on simulation optimization, and focuses on continuous optimization problems more than discrete optimization problems. The paper focuses specifically on discrete-event simulations. Fu [60] provides a comprehensive survey of the field and its scope—the paper outlines the different ways in which optimization and simulation interact, gives examples of real-world applications, introduces simulation software and the optimization routines that each of them use, provides a very basic tutorial on simulation output analysis and convergence theory for simulation optimization, elaborates on algorithms for both continuous and discrete problems, and provides pointers to many useful sources. Fu et al. [62] provide a concise, updated version of all of this, and also talk about estimation of distribution algorithms.

Tekin and Sabuncuoglu [194] provide a table that analyzes past review papers and the techniques they focus on. Apart from providing detailed updates on advances in approaches and algorithms, the paper also lists references that attempt to compare different SO techniques. Hong and Nelson [88] classify simulation optimization problems into those with (1) a finite number of solutions; (2) continuous decision variables; and (3) discrete variables that are integer-ordered. The paper describes procedures for each of these classes. Perhaps the most recent survey, [5], classifies simulation optimization algorithms and provides a survey of methods as well as applications appearing in the literature between 1995 and 2010.

The present paper provides an overview of techniques, and briefly outlines well-established methods with pointers to more detailed surveys, while expounding on more recent methods in a concise manner. Though several reviews exist, we catalog the most recent developments— the emergence of derivative-free optimization and its relationship with simulation optimization, the appearance of simulation test-beds for comparing algorithms, the recent application of simulation optimization in diverse fields, the development of and interest in related techniques and theory by the machine learning community and the optimization community, as well as the sheer unprecedented nature of recent interest in optimizing over simulations. A reflection of a surge in recent interest is evidenced by the fact that more than half of the works we reference were published in the last decade. The intent is to not only trace the progress of the field, but to provide an update on state-of-the-art methods and implementations, point the familiar as well as the uninitiated reader to relevant sources in the literature, and to speculate on future directions in the field.

1.2   A note on terminology and scope

As simulation optimization involves the use of algorithms that arose from widely differing fields (Section 3), has relationships to many diverse disciplines (Section 1.3), and has been applied to many different practical applications from biology to engineering to logistics (Section 2), it is not surprising that it is known by various names in different fields. It has also been referred to as simulation-based optimization, stochastic optimization, parametric optimization, black-box optimization, and Optimization via Simulation (OvS), where the continuous and discrete versions are accordingly known as Continuous Optimization via Simulation (COvS) and Discrete Optimization via Simulation (DOvS). Each algorithmic technique may also go by different names, and we attempt to reconcile these in Section 3.

Inputs to the simulation may be variously referred to as parameter settings, input settings, variables, controls, solutions, designs, experiments (or experimental designs), factors, or configurations. Outputs from the simulation are called measurements, responses, performance metrics, objective values, simulation replications, realizations, or results. The performance of a simulation may also be referred to as an experiment, an objective function evaluation, or simply a function evaluation. We will use the term 'iteration' to refer to a fixed number of function evaluations (usually one) performed by a simulation optimization

algorithm.

A note of caution while using SO methods is to incorporate as much domain specific knowledge as possible in the use of an SO algorithm. This may be in terms of (1) screening relevant input variables, (2) scaling and range reduction of decision variables, (3) providing good initial guesses for the algorithm; and (4) gleaning information from known problem structure, such as derivative estimates.

Table 1 classifies the techniques that are usually most suitable in practice for different scenarios in the universe of optimization problems. Certain broad classes of algorithms, such as random search methods, may be applicable to all of these types of problems, but they are often most suitable when dealing with pathological problems (e.g., problems with discontinuities, nonsmoothness) and are often used because they are relatively easy to implement.

The possibilities of combining simulation and optimization procedures are vast: simulation with optimization-based iterations; optimization with simulation-based iterations; sequential simulation and optimization; and alternate simulation and optimization are four such paradigms. A recent paper by Figueira and Almada-Lobo [56] delves into the taxonomy of such problems, and provides a guide to choosing an appropriate approach for a given problem. As detailed by Meketon [135], different techniques may be applicable or more suitable depending on how much is known about the underlying simulation, such as its structure or associated probability distributions. We focus on approaches that are applicable in situations where all the optimization scheme has to work with are evaluations of $f(x, y, \omega)$ and $g(x, y, \omega)$, or simply, observations with noise.

1.3 Relationship to other fields

Mathematical Programming. As mentioned earlier, most mathematical programming methods rely on the presence of an algebraic model. The availability of an algebraic model has many obvious implications to a mathematical programming expert, including the ability to evaluate a function quickly, the availability of derivative information, and the possibility of formulating a dual problem. None of these may be possible to do/obtain in an SO setting.

In the case with continuous decisions, derivative information is often hard to estimate accurately through finite differences, either due to the stochastic noise associated with objective function evaluations, or due to the large expense associated with obtaining function evaluations, or both. The inherent stochasticity in output also renders automatic differentiation (AD) [161, 76] tools not directly applicable. Moreover, automatic differentiation may not be used when one has no access to source code, does not possess an AD interface to proprietary simulation software, and, of course, when one is dealing with a physical experiment. The lack of availability of derivative information has further implications—it complicates the search for descent directions, proofs of convergence, and the characterization of optimal points.

Simulation optimization, like stochastic programming, also attempts to optimize under uncertainty. However, stochastic programming differs in that it makes heavy use of the model structure itself [27]. Optimization under uncertainty techniques that make heavy use of mathematical programming are reviewed in [175].

Derivative-Free Optimization. Both Simulation Optimization and Derivative-Free Optimization (DFO) are referred to in the literature as black-box optimization methods. Output variability is the key factor that distinguishes SO from DFO, where the output from the simulation is deterministic. However, there are many approaches to DFO that have analogs in SO as well (e.g., response surfaces, direct search methods, metaheuristics), cf. Section 3.

Another distinction is that most algorithms in DFO are specifically designed keeping in mind that function evaluations or simulations are expensive. This is not necessarily the case with SO algorithms.

With regard to rates of convergence, SO algorithms are generally inefficient and convergence rates are typically very slow. In general, one would expect SO to have a slower convergence rate than DFO algorithms simply because of the additional complication of uncertainty in function evaluations. As explained in [43], some DFO algorithms, under certain assumptions, expect rates that are closer to linear than quadratic, and therefore early termination may be suitable. As described in some detail by [59], the best possible convergence rates for SO algorithms are generally $\mathcal{O}(1/\sqrt{k})$, where k is the number of samples. This is true from the central limit theorem that tells us the rate at which the best possible estimator converges to the true expected function value at a point. This implies that though one would ideally incorporate rigorous termination criteria in algorithm implementations,

Table 1: Terminology of optimization problems

	Algebraic model available	Unknown/complex problem structure
Deterministic	Traditional math programming (linear, integer, and nonlinear programming)	Derivative-free optimization
Uncertainty present	Stochastic programming, robust optimization	Simulation optimization

most practical applications have a fixed simulation or function evaluation budget that is reached first.

Machine Learning. Several subcommunities in the machine learning community address problems closely related to simulation optimization. Traditional machine learning settings assume the availability of a fixed dataset. Active learning methods [40, 178] extend machine learning algorithms to the case where the algorithms are allowed to query an oracle for additional data to infer better statistical models. Active learning is closely related in that this choice of sampling occurs at every iteration in a simulation optimization setting as well. The focus of active learning is usually to learn better predictive models rather than to perform optimization.

Reinforcement learning [190] is broadly concerned with what set of actions to take in an environment to maximize some notion of cumulative reward. Reinforcement learning methods have strong connections to information theory, optimal control, and statistics. The similarity with simulation optimization is that the common problem of exploration of the search space vs. exploitation of known structure of the cost function arises. However, in the reinforcement learning setting, each action usually also incurs a cost, and the task is to maximize the accumulated rewards from all actions—as opposed to finding a good point in the parameter space eventually.

Policy gradient methods [155] are a subfield of reinforcement learning, where the set of all possible sequences of actions form the policy space, and a gradient in this policy space is estimated and a gradient ascent-type method is then used to move to a local optimum. Bandit optimization [68] is another subfield of reinforcement learning that involves methods for the solution to the multi-armed bandit problem. The canonical example involves a certain number of slot machines, and a certain total budget to play them. Here, each choice of sample corresponds to which slot machine to play. Each play on a slot machine results in random winnings. This setting is analogous to discrete simulation optimization (DOvS) over finite sets, although with a different objective [158]. Again, in DOvS over finite sets, we are only concerned with finding the best alternative eventually, whereas the cumulative winnings is the concern in the multi-armed bandit problem.

Relationship to other fields. Most, if not all, simulation optimization procedures have elements that are derived from or highly related to several other fields. Direct search procedures and response surface methodologies (RSM) have strong relationships with the field of experimental design. RSM, sample path optimization procedures, and gradient-based methods heavily incorporate ideas from mathematical programming. RSM also involves the use of nonparametric and Bayesian regression techniques, whereas estimation of distribution algorithms involves probabilistic inference, and therefore these techniques are related to statistics and machine learning. Simulation optimization has been described as being part of a larger field called computational stochastic optimization. More information is available at [157].

2   Applications

SO techniques are most commonly applied to either (1) discrete-event simulations, or (2) systems of stochastic nonlinear and/or differential equations.

As mentioned in [59], discrete event simulations can be used to model many real-world systems such as queues, operations, and networks. Here, the

simulation of a system usually involves switching or jumping from one state to another at discrete points in time as events occur. The occurrence of events is modeled using probability distributions to model the randomness involved.

Stochastic differential equations may be used to model phenomena ranging from financial risk [137] to the control of nonlinear systems [183] to the electrophoretic separation of DNA molecules [39].

With both discrete-event simulations and stochastic differential equation systems, there may be several parameters that one controls that affect some performance measure of the system under consideration, which are essentially degrees of freedom that may be optimized through SO techniques. Several applications of SO from diverse areas have been addressed in the literature and we list some of them in Table 2.

3 Algorithms

Algorithms for SO are diverse, and their applicability may be highly dependent on the particular application. For instance, algorithms may (1) attempt to find local or global solutions; (2) address discrete or continuous variables; (3) incorporate random elements or not; (4) be tailored for cases where function evaluations are expensive; (5) emphasize exploration or exploitation to different extents; (6) assume that the uncertainty in simulation output is homoscedastic or that it comes from a certain probability distribution; or (7) rely on underlying continuity or differentiability of the expectation (or some function of a chosen moment) of the simulation output. The sheer diversity of these algorithms also makes it somewhat difficult to assert which one is better than another in general, and also makes it hard to compare between algorithms or their implementations.

As mentioned in Section 1.3, many algorithms that are available for continuous simulation optimization have analogs in derivative-based optimization and in derivative-free optimization, where function evaluations are deterministic. In any case, the key lies in the statistics of how noise is handled, and how it is integrated into the optimization scheme. We will provide pointers to references that are applicable to simulation optimization in particular. A comprehensive review of methods for derivative-free optimization is available in [166].

Each major subsection below is accompanied by pseudocode to give researchers and practitioners unfamiliar with the field an idea of the general

approach taken by each of these algorithms. Many of the sections include pointers to convergence proofs for individual algorithms. Optimality in simulation optimization is harder to establish than in mathematical programming or derivative-free optimization due to the presence of output variability. Notions of optimality for simulation optimization are explored in [59]; for the discrete case, [206], for instance, establishes conditions for local convergence, where a point being 'better' than its $2m + 1$ neighboring solutions is said to be locally optimal. There has also been some work in establishing Karush-Kuhn-Tucker (KKT) optimality conditions for multiresponse simulation optimization [24]. Globally convergent algorithms will locate the global optimal solution eventually, but assuring this would require all feasible solutions to be evaluated through infinite observations; in practice, a convergence property that translates to a practical stopping criterion may make more sense [88].

Based on their scope, the broad classes of algorithms are classified in Table 3. Algorithms are classified based on whether they are applicable to problems with discrete/continuous variables, and whether they focus on global or local optimization. However, there may be specific algorithms that have been tweaked to make them applicable to a different class as well, which may not be captured by this table.

3.1 Discrete optimization via simulation

Discrete optimization via simulation is concerned with finding optimal settings for variables that can only take discrete values. This may be in the form of *integer-ordered* variables or *categorical* variables [153]. Integer-ordered variables are allowed to take on integer or discrete values within a finite interval, where the order of these values translates to some physical interpretation. For example, this could be the number of trucks available for vehicle routing, or the set of standard pipe diameters that are available for the construction of a manufacturing plant. Categorical variables refer to more general kinds of discrete decisions, ranging from conventional on-off (0-1 or binary) variables to more abstract decisions such as the sequence of actions to take given a finite set of actions. It should be noted that though integer-ordered variables, for instance, may be logically represented using binary variables, it may be beneficial to retain them as integer-ordered to exploit correlations in objective function values between adjacent integer values.

A rich literature in DOvS has developed over the last 50 years, and the specific methods

Table 2: Partial list of published works that apply simulation optimization

Domain of application	Application and citations
Operations	Buffer location [132], nurse scheduling [193], inventory management [120, 176], health care [11], queuing networks [63, 25, 138]
Manufacturing	PCB production [48], engine manufacturing [192], production planning [107, 114], manufacturing-cell design [98], kanban sizing [79]
Medicine and biology	Protein engineering [168], cardiovascular surgery [202], breast cancer epidemiology [55], bioprocess control [197, 165], ECG analysis [67], medical image analysis [136]
Engineering	Welded beam design [209], solid waste management [210], pollution source identification [14], chemical supply chains [103], antenna design [159], aerodynamic design [203, 205, 204, 122], distillation column optimization [162], well placement [18], servo system control [160], power systems [54], radar analysis [108]
Computer science, networks, electronics	Server assignment [125], wireless sensor networks [49], circuit design [130], network reliability [123]
Transportation and logistics	Traffic control and simulation [211, 17, 151], metro/transit travel times [83, 148], air traffic control [119, 96]

Table 3: Classification of simulation optimization algorithms

Algorithm class	Discrete	Continuous	Local	Global
Ranking and selection	×			×
Metaheuristics	×	×		×
Response surface methodology		×	×	×
Gradient-based methods		×	×	
Direct search	×	×	×	
Model-based methods	×	×	×	×
Lipschitzian optimization		×		×

developed are tailored to the specific problem setting. Broadly, methods are tailored for finite or for very large/potentially infinite parameter spaces.

3.1.1 Finite parameter spaces

In the finite case, where the number of alternatives is small and fixed, the primary goal is to decide how to allocate the simulation runs among the alternatives. In this setting, there is no emphasis on 'search', as the candidate solution pool is small and known; each iteration is used to infer the best, in some statistical sense, simulation run(s) to be performed subsequently.

The optimization that is desired may differ depending on the situation, and could involve:

1. The selection of the best candidate solution from a finite set of alternatives;

2. The comparison of simulation performance measures of each alternative to a known standard or control; or

3. The pairwise comparison between all solution candidates.

Item (1) is referred to as the *ranking and selection* problem. Items (2) and (3) are addressed under literature on *multiple comparison procedures*, with the former referred to as *multiple comparisons with a control*.

Ranking and Selection. In traditional ranking and selection, the task is to minimize the number of simulation replications while ensuring a certain probability of correct selection of alternatives. Most procedures try to guarantee that the design ultimately selected is better than all competing alternatives by δ with a probability at least $1 - \alpha$. δ is called the indifference zone, and is the value deemed to be sufficient to distinguish between expected performance among solution candidates. Conventional procedures make use of the Bonferroni inequality which relates probabilities of the occurrence of multiple events with probabilities of each event. Other approaches involve the incorporation of covariance induced by, for example, the use of common random numbers to expedite the algorithmic performance over the more conservative Bonferroni approach. Kim and Nelson [111, 112] and Chick [38] provide a detailed review and provide algorithms and procedures for this setting. Extensions of fully sequential ranking and selection procedures to the constrained case have been explored as well, e.g., [10].

An alternative formulation of the ranking and selection of the problem would be to try to do the best within a specified computational budget, called the *optimal computing budget allocation* formulation [33]. Chen et al. [36] present more recent work, while the stochastically constrained case is considered in [129].

Recent work [95] in the area of DOvS over finite sets provides a quick overview of the field of ranking and selection, and considers general probability distributions and the presence of stochastic constraints simultaneously.

A basic ranking and selection procedure [112] is outlined in Algorithm 1, where it is assumed that independent data comes from normal distributions with unknown, different variances.

Algorithm 1 Basic ranking and selection procedure for SO

Require: Confidence level $1-\alpha$, indifference zone parameter δ

1: Take n_0 samples from each of the $1, \ldots, K$ potential designs

2: Compute sample means, $\bar{t}_{k,n_0}$ and sample variances, S_k, for each of the designs

3: Determine how many new samples, $N_k := \max \left\{ n_0, \left\lceil \frac{\psi^2 S_k^2}{\delta^2} \right\rceil \right\}$, to take from each system, where the Rinott constant ψ is obtained from [21]

4: Select the system with the best new sample mean, $\bar{t}_{k,N_k+n_0}$.

Multiple comparison procedures. Here, a number of simulation replications are performed on all the potential designs, and conclusions are made by constructing confidence intervals on the performance metric. The main ideas and techniques for multiple comparisons in the context of pairwise comparisons, or against a known standard are presented in [86], [59], and [90]. Recent work in multiple comparisons with a control include [110] and [146], which provide fully sequential and two-stage frequentist procedures respectively; and [201], which addresses the problem using a Bayesian approach.

Comprehensive treatment of ranking and selection and multiple comparison procedures may be found in Goldsman and Nelson [74] and Bechhofer et al.

[21]. A detailed survey that traces the development of techniques in simulation optimization over finite sets is available in [194].

3.1.2 Large/Infinite parameter spaces

To address DOvS problems with a large number of potential alternatives, algorithms that have a search component are required. Many of the algorithms that are applicable to the continuous optimization via simulation case are, with suitable modifications, applicable to the case with large/infinite parameter spaces. These include (1) ordinal optimization (2) random search methods and (3) direct search methods.

Ordinal optimization methods [84] are suitable when the number of alternatives is too large to find the globally optimal design in the discrete-event simulation context. Instead, the task is to find a satisfactory solution with some guarantees on quality (called alignment probability) [127]. Here, the focus is on sampling a chosen subset of the solutions and evaluating them to determine the best among them. The key lies in choosing this subset such that it contains a subset of satisfactory solutions. The quality or satisfaction level of this selected subset can be quantified [34]. A comparison of subset selection rules is presented in [100] and the multi-objective case is treated in [195].

Random search methods include techniques such as simulated annealing (e.g., [3]), genetic algorithms, stochastic ruler methods (e.g., [208]), stochastic comparison (e.g., [75]), nested partitions (e.g., [182]), ant colony optimization (e.g., [52, 51]), and tabu search (e.g., [71]). Some of these— simulated annealing, genetic algorithms, and tabu search—are described in Section 3.6). Ant colony optimization is described under model-based methods (cf. Section 3.7.2). Proofs of global convergence, i.e., convergence to a global solution, or local convergence are available for most of these algorithms [88] (note that these definitions differ from mathematical programming where *global convergence* properties ensure convergence to a *local optimum* regardless of the starting point).

Nested partition methods [181] attempt to adaptively sample from the feasible region. The feasible region is then partitioned, and sampling is concentrated in regions adjudged to be the most promising by the algorithm from a pre-determined collection of nested sets. Hong and Nelson propose the COMPASS algorithm [87] which uses a unique neighborhood structure, defined as the most promising region that is fully adaptive

rather than pre-determined; a most promising 'index' is defined that classifies each candidate solution based on a nearest neighbor metric. More recently, the Adaptive Hyberbox Algorithm [207] claims to have superior performance on high-dimensional problems (problems with more than ten or fifteen variables); and the R-SPLINE algorithm [198], which alternates between a continuous search on a continuous piecewise-linear interpolation and a discrete neighborhood search, compares favorably as well.

A review of random search methods is presented in [8, 149]. Recent progress, outlines of basic algorithms, and pointers to specific references for some of these methods are presented in [26], [88], and [145].

Direct search methods such as pattern search and Nelder-Mead simplex methods are elaborated on in Section 3.5.

3.2 Response surface methodology

Response surface methodology (RSM) is typically useful in the context of continuous optimization problems and focuses on learning input-output relationships to approximate the underlying simulation by a surface (also known as a metamodel or surrogate model) for which we define a functional form. This functional form can then be made use of by leveraging powerful derivative-based optimization techniques. The literature in RSM is vast and equivalent approaches have variously been referred to as multi-disciplinary design optimization, metamodel-based optimization, and sequential parameter optimization. RSM was originally developed in the context of experimental design for physical processes [28], but has since been applied to computer experiments. Metamodel-based optimization is a currently popular technique for addressing simulation optimization problems [20, 115].

Different response surface algorithms differ in the choice between regression and interpolation; the nature of the functional form used for approximation (polynomials, splines, Kriging, radial basis functions, neural networks); the choice of how many and where new samples must be taken; and how they update the response surface.

RSM approaches can either (1) build surrogate models that are effective in local regions, and sequentially use these models to guide the search, or; (2) build surrogate models for the entire parameter space from space-filling designs, and then use them to choose samples in areas of interest, i.e., where the likelihood of finding better solutions is good

---

Algorithm 2 Basic RSM procedure

Require: Initial region of approximation $\mathcal{X}$, choice of regression surface r
1: **while** under simulation budget and not converged **do**
2:     Perform a design of experiments in relevant region, using k data points
3:     $t_i \leftarrow$ simulate(x_i), $\quad i = \{1, \ldots, k\}$ {Evaluate noisy function $f(x_i, \omega)$}
4:     $\lambda^* \leftarrow \arg\min_\lambda \sum (t_i - r(x_i, \lambda))^2$ {Fit regression surface r through points using squared loss function}
5:     $x^* \leftarrow \{\arg\min_x r(x, \lambda^*) : x \in \mathcal{X}\}$ {Optimize surface}
6:     Update set of available data points and region of approximation
7: **end while**

---

according to a specified metric. A generic framework for RSM is presented in Algorithm 2.

Classical sequential RSM. Originally, RSM consisted of a Phase I, where first-order models were built using samples from a design of experiments. A steepest descent rule was used to move in a certain direction, and this would continue iteratively until the estimated gradient would be close to zero. Then, a Phase II procedure that built a more detailed quadratic model would be used for verifying the optimality of the experimental design. A thorough introduction to response surface methodology is available in [142]. Recent work in the field includes automating RSM [143, 147] and the capability to handle stochastic constraints [13].

Bayesian global optimization. These methods seek to build a global response surface, commonly using techniques such as Kriging/Gaussian process regression [173, 163]. Subsequent samples chosen based on some sort of improvement metric may balance exploitation and exploration. The seminal paper by Jones et al. [102] which introduced the EGO algorithm for simulations with deterministic output, uses Kriging to interpolate between function values, and chooses future samples based on an expected improvement metric [140]. Examples of analogs to this for simulation optimization are provided in [93, 118]. The use of Kriging for

simulation metamodeling is explored in [22, 117, 116]. Other criteria that have been used to choose samples are most probable improvement [139], knowledge gradient for continuous parameters [177], and maximum information gain [189].

Trust region methods. Trust region methods [42] can be used to implement sequential RSM. Trust regions provide a means of controlling the region of approximation, providing update criteria for surrogate models, and are useful in analyzing convergence properties. Once a metamodel or response surface, g, is built around a trust region center x_i, trust region algorithms involve the solution of the trust-region subproblem $(\min_s g(x_i + s) : s \in \mathcal{B}(x_i, \Delta))$, where $\mathcal{B}$ is a ball defined by the center-radius pair (x_i, Δ). There are well-defined criteria to update the trust region center and radius [42] that will define the subsequent region of approximation.

The use of trust regions in simulation optimization is relatively recent, and has been investigated to some extent [46, 32]. Trust-region algorithms have been used, for example, to optimize simulations of urban traffic networks [151].

3.3 Gradient-based methods

Stochastic approximation methods or gradient-based approaches are those that attempt to descend using estimated gradient information. Stochastic approximation techniques are one of the oldest methods for simulation optimization. Robbins and Monro [167] and Kiefer and Wolfowitz [109] were the first to develop stochastic approximation schemes in the early 1950s. These procedures initially were meant to be used under very restrictive conditions, but much progress has been made since then.

These methods can be thought of being analogous to steepest descent methods in derivative-based optimization. One may obtain direct gradients or may estimate gradients using some finite difference scheme. Direct gradients may be calculated by a number of methods: (1) Perturbation Analysis (specifically, Infinitesimal Perturbation Analysis) (PA or IPA), (2) Likelihood Ratio/Score Function (LR/SF), and (3) Frequency Domain Analysis (FDA). Detailed books on these methods are available in the literature [85, 69, 172, 156, 65] and more high-level descriptions are available in papers [194, 60]. Most of these direct methods, however, are either applicable to specific kinds of problems, need some information about underlying distributions, or

are difficult to apply. Fu [60] outlines which methods are applicable in which situations, and Tekin and Sabuncuoglu [194] discuss a number of applications that have used these methods.

Stochastic approximation schemes attempt to estimate a gradient by means of finite differences. Typically, a forward difference estimate would involve sampling at least $n + 1$ distinct points, but superior performance has been observed by simultaneous perturbation estimates that require samples at just two points [185], a method referred to as Simultaneous Perturbation Stochastic Approximation (SPSA). The advantage gained in SPSA is that the samples required are now independent of the problem size, and, interestingly, this has been shown to have the same asymptotic convergence rate as the naive method that requires $n + 1$ points [184]. A typical gradient-based scheme is outlined in Algorithm 3.

Algorithm 3 Basic gradient-based procedure

Require: Specify initial point, x_0. Define initial parameters such as step size (α), distances between points for performing finite difference, etc.

1: $i \leftarrow 0$
2: **while** under simulation budget and not converged **do**
3:     Perform required simulations, $t_i^{j_i} \leftarrow$ simulate(x_i), with j_i replications to estimate gradient, $\widehat{J}$, using either IPA, LR/SF, FDA or finite differences
4:     $x_{i+1} \leftarrow x_i - \alpha \widehat{J}$
5:     $i \leftarrow i + 1$
6: **end while**

Recent extensions of the SPSA method include introducing a global search component to the algorithm by injecting Monte Carlo noise during the update step [134], and using it to solve combined discrete/continuous optimization problems [199]. Recent work also addresses improving Jacobian as well as Hessian estimates in the context of the SPSA algorithm [187]. Much of the progress in stochastic approximation has been cataloged in the proceedings of the Winter Simulation Conference over the years (http://informs-sim.org/). A recent review of stochastic approximation methods is available in [188], and an excellent tutorial and review

of results in stochastic approximation is presented in [154].

3.4 Sample path optimization

Sample path optimization involves working with an estimate of the underlying unknown function, as opposed to the function itself. The estimate is usually a consistent estimator such as the sample mean of independent function evaluations at a point, or replications. For instance, one may work with $F_n = \frac{1}{n} \sum_{i=1}^{n} f(x, y, \omega_i)$, instead of the underlying function $E[f(x, y, \omega)]$ itself. It should be noted that the functional form of F_n is still unknown, it is just that F_n can be observed or evaluated at a point in the search space visited by an algorithm iteration. The alternative name of sample average approximation reflects this use of an estimator.

As the algorithm now has to work with an estimator, a deterministic realization of the underlying stochastic function, sophisticated techniques from traditional mathematical programming can now be leveraged. Sample path methods can be viewed as the use of deterministic optimization techniques within a well-defined stochastic setting. Yet another name for them is stochastic counterpart. Some of the first papers using sample path methods are [82] and [179]. Several papers [172, 37, 77, 180, 46] discuss convergence results and algorithms in this context.

3.5 Direct search methods

Direct search can be defined as the sequential examination of trial solutions generated by a certain strategy [89]. As opposed to stochastic approximation, direct search methods rely on direct comparison of function values without attempting to approximate derivatives. Direct search methods typically rely on some sort of ranking of quality of points, rather than on function values.

Most direct search algorithms developed for simulation optimization are extensions of ideas for derivative-free optimization. A comprehensive review of classical and modern methods is provided in [121]. A formal theory of direct search methods for stochastic optimization is developed in [196]. Direct search methods can be tailored for both discrete and continuous optimization settings. Pattern search and Nelder-Mead simplex procedures are the most popular direct search methods. There is some classical as well as relatively recent work done on investigating both pattern search methods [196, 6, 131] and Nelder-Mead simplex algorithms [144, 19,

94, 31] and their convergence in the context of simulation optimization.

These methods remain attractive as they are relatively easy to describe and implement, and are not affected if a gradient does not exist everywhere, as they do not rely on gradient information. Since conventional procedures can be affected by noise, effective sampling schemes to control the noise are required. A basic Nelder-Mead procedure is outlined in Algorithm 4.

Algorithm 4 Basic Nelder-Mead simplex procedure for SO

Require: A set of $n - 1$ points in the parameter space to form the initial simplex

1: **while** under simulation budget and not converged **do**
2:     Generate a new candidate solution, x_i, through simplex centroid reflections, contractions or other means
3:     $t_i^{j_i} \leftarrow \text{simulate}(x_i), \quad i = \{i - n + 1, \ldots, i\}, j_i = \{1, \ldots, N_i\}$ {Evaluate noisy function $f(x, \omega)$ N_i times, where N_i is determined by some sampling scheme}
4:     Calculate $\frac{\sum_{j_i} t_i^{j_i}}{N_i}$, or some similar metric to determine which point (i.e., with the highest metric value) should be eliminated
5: **end while**

3.6 Random search methods

3.6.1 Genetic algorithms

Genetic algorithms use concepts of mutation and selection [164, 200]. In general, a genetic algorithm works by creating a population of strings and each of these strings are called chromosomes. Each of these chromosome strings is basically a vector of point in the search space. New chromosomes are created by using selection, mutation and crossover functions. The selection process is guided by evaluating the fitness (or objective function) of each chromosome and selecting the chromosomes according to their fitness values (using methods such as mapping onto Roulette Wheel). Additional chromosomes are then generated using crossover and mutation functions.

The cross over and mutation functions ensure that a diversity of solutions is maintained. Genetic algorithms are popular as they are easy to implement and are used in several commercial simulation optimization software packages (Table 4). The GECCO (Genetic and Evolutionary Computation Conference) catalogs progress in genetic algorithms and implementations.

3.6.2 Simulated annealing

Simulated Annealing uses a probabilistic method that is derived from the annealing process in which the material is slowly cooled so that, while its structure freezes, it reaches a minimum energy state [113, 23]. Starting with a current point i in a state j, a neighborhood point i' of the point i is generated. The algorithm moves from point i to i' using a probabilistic criteria that is dependent on the 'temperature' in state j. This temperature is analogous to that in physical annealing, and serves here as a control parameter. If the solution at i' is better than the existing solution, then this new point is accepted. If the new solution is worse than existing solution, then the probability of accepting the point is defined as $\exp(-(f(i') - f(i))/T(j))$, where $f(.)$ is the value of objective function at a given point, and $T(j)$ is temperature at the state j. After a certain number of neighborhood points are evaluated, the temperature is decreased and new state is $j + 1$ is created. Due to the exponential form, the probability of acceptance of a neighborhood point is higher at high temperature, and is lower as temperature is reduced. In this way, the algorithm searches for a large number of neighborhood points in the beginning, but a lower number of points as temperature is reduced.

Implementation of simulated annealing procedures require choosing parameters such as the initial and final temperatures, the rate of cooling, and number of function evaluations at each temperature. A variety of cooling 'schedules' have been suggested in [41] and [78]. Though simulated annealing was originally meant for optimizing deterministic functions, the framework has been extended to the case of stochastic simulations [2]. The ease of implementing a simulated annealing procedure is high and it remains a popular technique used by several commercial simulation optimization packages.

3.6.3 Tabu search

Tabu search [70] uses special memory structures (short-term and long-term) during the search

process that allow the method to go beyond local optimality to explore promising regions of the search space. The basic form of tabu search consists of a modified neighborhood search procedure that employs adaptive memory to keep track of relevant solution history, together with strategies for exploiting this memory [66]. More advanced forms of tabu search and its applications are described in [72].

3.6.4 Scatter search

Scatter search and its generalized form, path relinking, were originally introduced by Glover and Laguna [73]. Scatter search differs from other evolutionary approaches (such as Genetic Algorithms (GA)) by using strategic designs and search path construction from a population of solutions as compared to randomization (by crossover and mutation in GA). Similar to Tabu search, Scatter Search also utilize adaptive memory in storing best solutions [73, 133]. Algorithm 5 provides the scatter search algorithm.

3.7 Model-based methods

Model-based simulation optimization methods attempt to build a probability distribution over the space of solutions and use it to guide the search process.

3.7.1 Estimation of distribution algorithms

Estimation of distribution algorithms (EDAs) [126] are model-based methods that belong to the evolutionary computation field. However, generation of new candidate solutions is done by sampling from the inferred probability distribution over the space of solutions, rather than, say, a genetic operator such as crossover or mutation. A comprehensive review of estimation of distribution algorithms is presented in [64]. EDAs usually consider interactions between the problem variables and exploit them through different probability models.

Cross-entropy methods and Model Reference Adaptive Search (MRAS) are discussed next and can be seen as specific instances of EDAs.

Cross-Entropy Methods. Cross-entropy methods first sample randomly from a chosen probability distribution over the space of decision variables. For each sample, which is a vector defining a point in decision space, a corresponding function evaluation is obtained. Based on the function values

---

Algorithm 5 Basic scatter search procedure for SO

Require: An initial set of trial points $x \in P$, chosen to be diversified according to a pre-specified metric

1: $t_j \leftarrow \text{simulate}(x_j)$, where $j = 1, \ldots, |P|$
2: $k \leftarrow 0$
3: Use a comparison procedure (such as ranking and selection) to gather the best b solutions (based on objective value or diversity) from the current set of solutions P, called the reference set, R_k
4: $R_{-1} = \varnothing$
5: **while** under simulation budget and $R_k \neq R_{k-1}$ **do**
6: $\quad k \leftarrow k + 1$
7: $\quad$ Choose $S_i \subset R$, where $i = 1, \ldots, r$ {Use a subset generation procedure to select r subsets of set R, to be used as a basis for generating new solution points}
8: $\quad$ **for** $i = 1$ to r **do**
9: $\quad\quad$ Combine the points in S_i, to form new solution points, x_j, where $j \in \mathcal{J} = |P| + 1, \ldots, |P| + J$, using weighted linear combinations, for example
10: $\quad\quad$ $t_j \leftarrow \text{simulate}(x_j)$, $j \in \mathcal{J}$ {sample the objective function at new trial solutions}
11: $\quad\quad$ Update sets R_k, P
12: $\quad$ **end for**
13: **end while**

observed, a pre-defined percentile of the best samples are picked. A new distribution is built around this 'elite set' of points via maximum likelihood estimation or some other fitting method, and the process is repeated. One possible method that implements cross-entropy is formally described in Algorithm 6.

Algorithm 6 Pseudocode for a simple cross-entropy implementation

---

Require: θ, an initial set of parameters for a pre-chosen distribution $p(x; \theta)$ over the set of decision variables; k, a number of simulations to be performed; e, the number of elite samples representing the top δ percentile of the k samples
1: **while** under simulation budget and not converged **do**
2:     **for** $i = 1 \rightarrow k$ **do**
3:         sample x_i from $p(x; \theta)$
4:         $t_i \leftarrow \text{simulate}(x_i)$
5:     **end for**
6:     $E \leftarrow \emptyset$
7:     **for** $i = 1 \rightarrow e$ **do**
8:         $E_j \leftarrow \arg\max_{i \notin E} t_i$
9:     **end for**
10:    $p(x; \theta) \leftarrow \text{fit}(x_E)$
11: **end while**

The method is guaranteed (probabilistically) to converge to a local optimum, but it also incorporates an exploration component as random samples are obtained at each step. However, the intuition behind the selection of subsequent samples can be shown to be analogous to minimizing the Kullback-Leibler divergence (KL-divergence) between the optimal importance sampling distribution and the distribution used in the current iterate [171].

There exist variants of the cross-entropy method to address both continuous [124] and discrete optimization [170] problems. A possible modification is to use mixtures of distributions from current and previous iterations, with the current distribution weighted higher. This can be done by linearly interpolating the mean covariance in the case of Gaussian distributions. This also helps in avoiding singular covariance matrices. Cross-entropy can also deal with noisy function evaluations, with irrelevant decision variables, and constraints [124]. If

decision variables are correlated, the covariance of the distribution will reflect this.

The immediately apparent merits of cross-entropy methods are that they are easy to implement, require few algorithmic parameters, are based on fundamental principles such as KL-divergence and maximum likelihood, and give consistently accurate results [124]. A potential drawback is that cross-entropy may require a significant number of new samples at every iteration. It is not clear as to how this would affect performance if samples were expensive to obtain. The cross-entropy method has analogs in simulated annealing, genetic algorithms, and ant colony optimization, but differs from each of these in important ways [44].

More detailed information on the use of cross-entropy methods for optimization can be found in [44], a tutorial on cross-entropy and in [171], a monograph. The cross-entropy webpage, http://iew3.technion.ac.il/CE/ provides up-to-date information on progress in the field.

Model reference adaptive search (MRAS). The MRAS method [91, 92] is closely related to the cross-entropy method. It also works by minimizing the Kullback-Leibler divergence to update the parameters of the inferred probability distribution. However, the parameter update step involves the use of a sequence of implicit probability distributions. In other words, while the cross-entropy method uses the optimal importance sampling distribution for parameter updates, MRAS minimizes the KL-divergence with respect to the distribution in the current iteration, called the reference model.

Covariance Matrix Adaptation–Evolution Strategy (CMA-ES). In the CMA-ES algorithm [81], new samples are obtained from a multivariate normal distribution, and inter-variable dependencies are encoded in the covariance matrix. The CMA-ES method provides a way to update the covariance matrix. Updating the covariance matrix is analogous to learning an approximate inverse Hessian, as is used in Quasi-Newton methods in mathematical programming. The update of the mean and covariance is done by maximizing the likelihood of previously successful candidate solutions and search steps, respectively. This is in contrast to other EDAs and the cross-entropy method, where the covariance is updated by maximizing the likelihood of the successful points. Other sophistications such as step-size control, and weighting of candidate

solutions are part of modern implementations [80].

3.7.2   Ant colony optimization

Ant colony optimization methods [52, 51] are heuristic methods that have been used for combinatorial optimization problems. Conceptually, they mimic the behavior of ants to find shortest paths between their colony and food sources. Ants deposit pheromones as they walk; and are more likely to choose paths with higher concentration of pheromones. This phenomenon is incorporated in a pheromone update rule, which increases the pheromone content in components of high-quality solutions, and causes evaporation of pheromones in less favorable regions. Probability distributions are used to make the transition between each iteration. These methods differ from EDAs in that they use an iterative construction of solutions.

This and other algorithms that incorporate self-organization in biological systems are said to use the concept of 'swarm intelligence'.

3.8   Lipschitzian optimization

Lipschitzian optimization is a class of space-partitioning algorithms for performing global optimization, where the Lipschitz constant is pre-specified. This enables the construction of global search algorithms with convergence guarantees. The caveat of having prior knowledge of the Lipschitz constant is overcome by the DIRECT (DIviding RECTangles) algorithm [101] for deterministic continuous optimization problems. An adaptation of this for noisy problems is provided in [47].

4   Software

4.1   Simulation optimization in commercial simulation software

Many discrete-event simulation packages incorporate some methodology for performing optimization. A comprehensive listing of simulation software, the corresponding vendors, and the optimization packages and techniques they use can be found in Table 4. More details on the specific optimization routines can be found in [128]. OR/MS-Today, the online magazine of INFORMS, conducts a biennial survey of simulation software packages, the latest of which is available at [150]. The survey lists 43 simulation software packages, and 31 of these

have some sort of optimization routine; fewer still have black-box optimizers that interact with the simulation.

4.2   Academic implementations of simulation optimization

Table 5 contains a small subset of academic implementations of SO algorithms, and classifies them by type. Some of these are available for download from the web, some have code with suggested parameters in corresponding papers themselves, and others are available upon request from the authors.

5   Comparison of algorithms

As far as comparisons between algorithms are concerned, the literature does not yet provide a comprehensive survey of the performance of different implementations and approaches on large test beds. In this regard, simulation optimization lags behind other optimization fields such as linear, integer, and nonlinear programming, global optimization and even derivative-free optimization, where the first comprehensive comparison appeared in 2013 [166]. A study of prior comparisons in simulation optimization is provided by [194], but these comparisons are fairly dated, are inconclusive about which algorithms perform better in different situations, and compare only a small subset of available algorithms. One difficulty lies in the inherent difficulty of comparing solutions between algorithms over true black-box simulations, as one does not usually know the true optimal point and can only compare between noisy estimates observed by the solvers. Less impeding difficulties, but difficulties nonetheless, include the need to interface algorithms to a common wrapper, the objective comparison with solvers that incorporate random elements as their results may not be reproducible, and lack of standard test simulations for purposes of benchmarking.

The benchmarking of algorithms in mathematical programming is usually done by performance profiles [50], where the graphs show the fraction of problems solved after a certain time. For derivative-free algorithms, data profiles are commonly used [141], where the fraction of problems solved after a certain number of iterations (function evaluations) or 'simplex gradients' is shown. The definition of when a problem is 'solved' may vary—when the true global optimum is known, the solutions found within a certain tolerance of this optimal value may be called

Table 4: Simulation optimization packages in commercial simulation software

Optimization package	Vendor	Simulation software supported	Optimization methodology
AutoStat	Applied Materials, Inc.	AutoMod	Evolutionary strategy
Evolutionary Optimizer	Imagine That, Inc.	ExtendSim	Evolutionary strategy
OptQuest	OptTek Systems, Inc.	FlexSim, @RISK, Simul8, Simio, SIMPROCESS, AnyLogic, Arena, Crystal Ball, Enterprise Dynamics, ModelRisk	Scatter search, tabu search, neural networks, integer programming
SimRunner	ProModel Corp.	ProModel, MedModel, ServiceModel	Genetic algorithms and evolutionary strategies
RISKOptimizer	Palisade Corp.	@RISK	Genetic algorithm
WITNESS Optimizer	Lanner Group, Inc.	WITNESS	Simulated annealing, tabu search, hill climbing
GoldSim Optimizer	GoldSim Technology Group	GoldSim	Box's complex method
Plant Simulation Optimizer	Siemens AG	Siemens PLM software	Genetic algorithm
ChaStrobeGA	N/A	Stroboscope	Genetic algorithm
Global Optimization toolbox	The MathWorks	SimEvents (Matlab)	Genetic algorithms, simulated annealing, pattern search

Table 5: Academic simulation optimization implementations

Algorithm	Type	Citation	Year
Continuous			
SPSA	Stochastic approximation	[185]	1992
SPSA 2nd Order	Stochastic approximation	[185]	1999
SKO	Global response surface	[93]	2006
CE method	Cross-entropy	[124]	2006
APS	Nested partitioning	[105]	2007
SNOBFIT	Multi-start local response surface	[97]	2008
CMA-ES	Evolutionary strategy	[80]	2011
KGCP	Global response surface	[177]	2011
STRONG	Local response surface, trust region	[32]	2011
GR	Golden region search	[106]	2011
SNM	Direct search (Nelder-Mead)	[31]	2012
DiceOptim	Global response surface	[169]	2012
Discrete			
KG	Global response surface	[57]	2009
COMPASS	Neighborhood search (integer-ordered problems)	[206]	2010
R-SPLINE	Neighborhood search (integer-ordered problems)	[198]	2012
Discrete and continuous			
MRAS	Estimation of distribution	[91, 92]	2005
NOMADm	Mesh adaptive direct search	[1]	2007

solutions, but when this optimum is not known, the solvers that find the best solution (within a tolerance) for a problem, with respect to the other solvers being compared, may be said to have solved the problem. The latter metric may also be used when function evaluations are expensive, and no solver is able to reach within this tolerance given the limited simulation budget.

In both of these cases, the output of the simulations are deterministic, and so it is clear as to which algorithms have performed better than others on a particular problem. In simulation optimization, however, usually one does not know the true solution for the black box system, nor does one see deterministic output. All that one possesses are mean values and sample variances obtained from sample paths at different points. There does not exist a standard method to compare simulation optimization algorithms on large test beds. Many papers perform several macroreplications and report the macroreplicate average of the best sample means (along with the associated sample variance) at the end of the simulation budget. The issue with this is that the performance of the algorithms with different simulation budgets is not seen, as in the case of performance or data profiles. Other papers report the average number of evaluations taken to find a sample mean that is within the global tolerance for each problem. Here, results are listed for each problem and one does not get an idea of overall performance. In addition, the difference in sample variance estimates is not highlighted. As simulation optimization develops, there is also a need for methods of comparison of algorithms on test beds

with statistically significant number of problems.

With regard to standardized simulation testbeds, to our knowledge, the only testbed that provides practical simulations for testing simulation optimization algorithms is available at www.simopt.org [153]. At the point of writing this paper, just 20 continuous optimization problems were available from this repository. Most testing and comparisons happen with classical test problems in nonlinear optimization (many of which have been compiled in [166] and available at http://archimedes.cheme.cmu.edu/?q=dfocomp), to which stochastic noise has been added. There is a need for more such repositories, not only for testing of algorithms over statistically significant sizes of problem sets, but for comparison between different classes of algorithms. The need for comparison is evident, given the sheer number of available approaches to solving simulation optimization problems, and the lack of clarity and lack of consensus on which types of algorithms are suitable in which contexts.

As observed by several papers [61, 194, 88], there continues to exist a significant gap between research and practice in terms of algorithmic approaches. Optimizers bundled with simulation software, as observed in Section 4, tend to make use of algorithms which seem to work well but do not come with provable statistical properties or guarantees of local or global convergence. Academic papers, on the other hand, emphasize methods that are more sophisticated and prove convergence properties. One reason that may contribute to this is that very few simulation optimization algorithms arising from the research community are easily accessible. We wholeheartedly encourage researchers to post their executable files, if not their source code. This could not only encourage practitioners to use these techniques in practice, but allow for comparisons between methods and the development of standardized interfaces between simulations and simulation optimization software.

6   Conclusions

The field of simulation optimization has progressed significantly in the last decade, with several new algorithms, implementations, and applications. Contributions to the field arise from researchers and practitioners in the industrial engineering/operations research, mathematical programming, statistics and machine learning, as well as the computer science communities. The use of simulation to model complex, dynamic, and stochastic systems has only increased with computing power and availability of a wide variety of simulation languages. This increased use is reflected in the identification and application of simulation and simulation optimization methods to diverse fields in science, engineering, and business. There also exist strong analogies between, and ideas that may be borrowed from recent progress in related fields. All of these factors, along with the ever increasing number of publications and rich literature in this area, clearly indicate the interest in the field of simulation optimization, and we have tried to capture this in this paper.

With increased growth and interest in the field, there come also opportunities. Potential directions for the field of simulation optimization are almost immediately apparent. Apart from the ability to handle simulation outputs from any well-defined probability distribution, the effective use of variance reduction techniques when possible, and the improvement in theory and algorithms, there is a requirement to address (1) large-scale problems with combined discrete/continuous variables; (2) the ability to effectively handle stochastic and deterministic constraints of various kinds; (2) the effective utilization of parallel computing at the linear algebra level, sample replication level, iteration level, as well as at the algorithmic level; (3) the effective handling of multiple simulation outputs; (4) the incorporation of performance measures other than expected values, such as risk; (5) the continued consolidation of various techniques and their potential synergy in hybrid algorithms; (6) the use of automatic differentiation techniques in the estimation of simulation derivatives when possible; (7) the continued emphasis on providing guarantees of convergence to optima for local and global optimization routines in general settings; (8) the availability and ease of comparison of the performance of available approaches on different applications; and (9) the continued reflection of sophisticated methodology arising from the literature in commercial simulation packages.

References

[1] Abramson, M.A.: NOMADm version 4.5 User's Guide. Air Force Institute of Technology, Wright-Patterson AFB, OH (2007)

[2] Alkhamis, T.M., Ahmed, M.A., Tuan, V.K.: Simulated annealing for discrete optimization with estimation. European Journal of Operational Research **116**, 530–544 (1999)

[3] Alrefaei, M.H., Andradóttir, S.: A simulated annealing algorithm with constant temperature for discrete stochastic optimization. Management Science **45**, 748–764 (1999)

[4] Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J.: Simulation optimization: A review of algorithms and applications. 4OR–A Quarterly Journal of Operations Research **12**, 301–333 (2014)

[5] Ammeri, A., Hachicha, W., Chabchoub, H., Masmoudi, F.: A comprehensive literature review of mono-objective simulation optimization methods. Advances in Production Engineering & Management **6**(4), 291–302 (2011)

[6] Anderson, E.J., Ferris, M.C.: A direct search algorithm for optimization with noisy function evaluations. SIAM Journal on Optimization **11**, 837–857 (2001)

[7] Andradóttir, S.: Chapter 9: Simulation optimization. In: J. Banks (ed.) Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons, New York (1998)

[8] Andradóttir, S.: An overview of simulation optimization via random search. In: S.G. Henderson, B.L. Nelson (eds.) Handbooks in Operations Research and Management Science: Simulation, vol. 13, chap. 20, pp. 617–631. Elsevier (2006)

[9] Andradóttir, S.: Simulation optimization. In: Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, pp. 307–333. John Wiley & Sons (2006)

[10] Andradóttir, S., Kim, S.H.: Fully sequential procedures for comparing constrained systems via simulation. Naval Research Logistics **57**(5), 403–421 (2010)

[11] de Angelis, V., Felici, G., Impelluso, P.: Integrating simulation and optimisation in health care centre management. European Journal of Operational Research **150**, 101–114 (2003)

[12] Angün, E.: Black box simulation optimization: Generalized response surface methodology. Ph.D. thesis, Tilburg University (2004)

[13] Angün, E., Kleijnen, J.P.C., Hertog, D.D., Gurkan, G.: Response surface methodology with stochastic constraints for expensive simulation. Journal of the Operational Research Society **60**(6), 735–746 (2009)

[14] Ayvaz, M.T.: A linked simulation-optimization model for solving the unknown groundwater pollution source identification problems. Journal of Contaminant Hydrology **117**(1–4), 46–59 (2010)

[15] Azadivar, F.: A tutorial on simulation optimization. In: J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson (eds.) Proceedings of the 1992 Winter Simulation Conference, pp. 198–204 (1992)

[16] Azadivar, J.: Simulation optimization methodologies. In: P.A. Farrington, H.B. Nembhard, D.T. Sturrock, G.W. Evans (eds.) Proceedings of the 1999 Winter Simulation Conference, pp. 93–100 (1999)

[17] Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., Wen, Y.: Calibration of microscopic traffic simulation models: Methods and application. Transportation Research Record: Journal of the Transportation Research Board **1999**(1), 198–207 (2007)

[18] Bangerth, W., Klie, H., Matossian, V., Parashar, M., Wheeler, M.F.: An autonomic reservoir framework for the stochastic optimization of well placement. Cluster Computing **8**(4), 255–269 (2005)

[19] Barton, R.R., Ivey, Jr., J.S.: Nelder-Mead simplex modifications for simulation optimization. Management Science **42**, 954–973 (1996)

[20] Barton, R.R., Meckesheimer, M.: Metamodel-based simulation optimization. In: S. Henderson, B. Nelson (eds.) Handbook in Operations Research and Management Science: Simulation 13, pp. 535–574. Elsevier (2006)

[21] Bechhofer, R.E., Santner, T.J., Goldsman, D.M.: Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons. John Wiley & Sons (1995)

[22] van Beers, A.C., Kleijnen, J.P.C.: Kriging interpolation in simulation: A survey. Proceedings of the 2004 Winter Simulation Conference **1**, 121–129 (2004)

[23] Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statistical Science **8**(1), 10–15 (1993)

[24] Bettonvil, B., del Castillo, E., Kleijnen, J.P.C.: Statistical testing of optimality conditions in multiresponse simulation-based optimization. European Journal of Operational Research **199**, 448–458 (2009)

[25] Bhatnagar, S.: Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. ACM Transactions on Modeling and Computer Simulation **15**(1), 74–107 (2005)

[26] Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J.: A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing **8**(2), 239–287 (2009)

[27] Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming, 2nd edn. Springer (2011)

[28] Box, G.E.P., Wilson, K.B.: On the experimental attainment of optimum conditions. Journal of the Royal Statistical Society **XIII**(1), 1–35 (1951)

[29] Carson, Y., Maria, A.: Simulation optimization: Methods and applications. In: S. Andradóttir, K.J. Healy, D.H. Winters, B.L. Nelson (eds.) Proceedings of the 1997 Winter Simulation Conference, pp. 118–126 (1997)

[30] Chang, K.H.: Stochastic trust region response surface convergent method for continuous simulation optimization. Ph.D. thesis, Purdue University (2008)

[31] Chang, K.H.: Stochastic Nelder-Mead simplex method–A new globally convergent direct search method for simulation optimization. European Journal of Operational Research **220**, 684–694 (2012)

[32] Chang, K.H., Hong, L.J., Wan, H.: Stochastic trust-region response-surface method (STRONG)—A new response-surface framework for simulation optimization. INFORMS Journal on Computing **25**(2), 230–243 (2013)

[33] Chen, C.H.: An effective approach to smartly allocate computing budget for discrete event simulation. In: Proceedings of the 34th IEEE Conference on Decision and Control, pp. 2598–2605 (1995)

[34] Chen, C.H.: A lower bound for the correct subset selection probability and its application to discrete event system simulations. IEEE Transactions on Automatic Control **41**, 1227–1231 (1996)

[35] Chen, C.H., Lee, L.H.: Stochastic Simulation Optimization: An Optimal Computing Budget Allocation. System Engineering and Operations Research. World Scientific Publishing Company (2010)

[36] Chen, C.H., Yücesan, E., Dai, L., Chen, H.C.: Optimal budget allocation for discrete-event simulation experiments. IIE Transactions **42**(1), 60–70 (2009)

[37] Chen, H., Schmeiser, B.W.: Retrospective optimization algorithms for stochastic root finding. In: J. Tew, S. Manivannan, D. Sadowski, A. Seila (eds.) Proceedings of 1994 Winter Simulation Conference, pp. 255–261 (1994)

[38] Chick, S.E.: Subjective probability and bayesian methodology. In: S.G. Henderson, B.L. Nelson (eds.) Simulation, *Handbooks in Operations Research and Management Science*, vol. 13, pp. 225–257. Elsevier, Amsterdam (2006)

[39] Cho, J., Dorfman, K.D.: Brownian dynamics simulations of electrophoretic DNA separations in a sparse ordered post array. Journal of Chromatography A **1217**, 5522–5528 (2010)

[40] Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. Journal of Artificial Intelligence Research **4**, 129–145 (1996)

[41] Collins, N.E., Eglese, R.W., Golden, B.L.: Simulated annealing–an annotated bibliography. American Journal of Mathematical Management Sciences **8**, 209–308 (1988)

[42] Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-region methods. MOS-SIAM Series on Optimization (2000)

[43] Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to derivative-free optimization. SIAM, Philadelphia, PA (2009)

[44] de Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annals of Operations Research **134**, 19–67 (2005)

[45] Deng, G.: Simulation-based optimization. Ph.D. thesis, University of Wisconsin–Madison (2007)

[46] Deng, G., Ferris, M.C.: Adaptation of the UOBYQA algorithm for noisy functions. In: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto (eds.) Proceedings of the 2006 Winter Simulation Conference, pp. 312–319 (2006)

[47] Deng, G., Ferris, M.C.: Extension of the DIRECT optimization algorithm for noisy functions. In: S.G. Henderson, B. Biller, M.H. Hsieh, J. Shortle, J.D. Tew, R.R. Barton (eds.) Proceedings of the 2007 Winter Simulation Conference, pp. 497–504 (2007)

[48] Dengiz, B., Akbay, K.S.: Computer simulation of a PCB production line: Metamodeling approach. International Journal of Production Economics **63**(2), 195–205 (2000)

[49] Dhivya, M., Sundarambal, M., Anand, L.N.: Energy efficient computation of data fusion in wireless sensor networks using cuckoo-based particle approach (cbpa). International Journal of Communications, Network and System Sciences **4**(4), 249–255 (2011)

[50] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**, 201–213 (2002)

[51] Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science **344**(2–3), 243–278 (2005)

[52] Dorigo, M., Stützle, T.: Ant colony optimization. MIT Press, Cambridge, MA, USA (2004)

[53] Driessen, L.T.: Simulation-based optimization for product and process design. Ph.D. thesis, Tilburg University (2006)

[54] Ernst, D., Glavic, M., Stan, G.B., Mannor, S., Wehenkel, L.: The cross-entropy method for power system combinatorial optimization problems. In: Power Tech, pp. 1290–1295. IEEE (2007)

[55] Ferris, M.C., Deng, G., Fryback, D.G., Kuruchittham, V.: Breast cancer epidemiology: Calibrating simulations via optimization. Oberwolfach Reports **2** (2005)

[56] Figueira, G., Almada-Lobo, B.: Hybrid simulation-optimization methods: A taxonomy. Simulation Modelling Practice and Theory **46**, 118–134 (2014)

[57] Frazier, P., Powell, W., Dayanik, S.: The knowledge-gradient policy for correlated normal beliefs. INFORMS Journal on Computing **21**(4), 599–613 (2009)

[58] Frazier, P.I.: Knowledge-gradient methods for statistical learning. Ph.D. thesis, Princeton University (2009)

[59] Fu, M.C.: Optimization via simulation: A review. Annals of Operations Research **53**, 199–247 (1994)

[60] Fu, M.C.: Optimization for simulation: Theory vs. practice. SIAM Journal on Computing **14**(3), 192–215 (2002)

[61] Fu, M.C., Andradóttir, S., Carson, J.S., Glover, F.W., Harrell, C.R., Ho, Y.C., Kelly, J.P., Robinson, S.M.: Integrating optimization and simulation: Research and practice. In: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (eds.) Proceedings of the 2000 Winter Simulation Conference (2000)

[62] Fu, M.C., Glover, F.W., April, J.: Simulation Optimization: A review, new developments, and applications. In: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, J.A. Joines (eds.) Proceedings of the 2005 Winter Simulation Conference, pp. 83–95 (2005)

[63] Fu, M.C., Hill, S.D.: Optimization of discrete event systems via simulataneous perturbation stochastic approximation. IIE Transactions **29**(233–243) (1997)

[64] Fu, M.C., Hu, J., Marcus, S.I.: Model-based randomized methods for global optimization. In: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, pp. 355–363. Kyoto, Japan (1996)

[65] Fu, M.C., Hu, J.Q.: Conditional Monte Carlo: Gradient estimation and optimization applications. Kluwer Academic Publishers (1997)

[66] Gendreau, M., Potvin, J.Y.: Tabu search. In: Handbook of Metaheuristics, *International Series in Operations Research & Management Science*, vol. 146, second edn., pp. 41–60. Springer (2010)

[67] Gerencsér, L., Kozmann, G., Vágó, Z., Haraszti, K.: The use of the spsa method in ecg analysis. IEEE Transactions on Biomedical Engineering **49**(10), 1094–1101 (2002)

[68] Gittins, J.C.: Multi-armed bandit allocation indices. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons (1989)

[69] Glasserman, P.: Gradient Estimation via Perturbation Analysis. Kluwer Academic Publishers (1991)

[70] Glover, F.: Tabu search: A tutorial. Interfaces **20**(4), 77–94 (1990)

[71] Glover, F., Hanafi, S.: Tabu search and finite convergence. Discrete Applied Mathematics **119**(1–2), 3–36 (2002)

[72] Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston (1997)

[73] Glover, F., Laguna, M.: Fundamentals of scatter search and path relinking. Control and Cybernetics **29**(3), 653–684 (2000)

[74] Goldsman, D., Nelson, B.L.: Comparing systems via simulation. In: J. Banks (ed.) Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice, chap. 8. John Wiley & Sons (1998)

[75] Gong, W.B., Ho, Y.C., Zhai, W.: Stochastic comparison algorithm for discrete optimization with estimation. SIAM Journal on Optimization **10**, 384–404 (49)

[76] Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, 2nd edn. No. 105 in Other Titles in Applied Mathematics. SIAM, Philadelphia, PA (2008). URL http://www.ec-securehost.com/SIAM/OT105.html

[77] Gürkan, G., Ozge, A.Y., Robinson, S.M.: Sample path optimization in simulation. In: J. Tew, S. Manivannan, D. Sadowski, A. Seila (eds.) Proceedings of 1994 Winter Simulation Conference, pp. 247–254 (1994)

[78] Hajek, B.: Cooling schedules for optimal annealing. Mathematics of Operations Research **13**, 311–329 (1988)

[79] Hall, J.D., Bowden, R.O., Usher, J.M.: Using evolution strategies and simulation to optimize a pull production system. Journal of Materials Processing Technology **61**(1–2), 47–52 (1996)

[80] Hansen, N.: The CMA Evolution Strategy: A tutorial. http://www.lri.fr/~hansen/cmaesintro.html

[81] Hansen, N.: The CMA evolution strategy: A comparing review. In: J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (eds.) Towards a new evolutionary computation. Advances on estimation of distribution algorithms, pp. 75–102. Springer (2006)

[82] Healy, K., Schruben, L.W.: Retrospective simulation response optimization. In: B.L. Nelson, D.W. Kelton, G.M. Clark (eds.) Proceedings of the 1991 Winter Simulation Conference, pp. 954–957 (1991)

[83] Hill, S.D., Fu, M.C.: Transfer optimization via simultaneous perturbation stochastic approximation. In: C. Alexopoulos, K. Kang, W.R. Lilegdon, D. Goldsman (eds.) Proceedings of the 1995 Winter Simulation Conference, pp. 242–249 (1995)

[84] Ho, Y.C.: An explanation of ordinal optimization: Soft computing for hard problems. Information Sciences **113**, 169–192 (1999)

[85] Ho, Y.C., Cao, X.R.: Discrete Event Dynamic Systems and Perturbation Analysis. Kluwer Academic Publishers (1991)

[86] Hochberg, Y., Tamhane, A.C.: Multiple Comparison Procedures. John Wiley & Sons, New York (1987)

[87] Hong, L.J., Nelson, B.L.: Discrete optimization via simulation using COMPASS. Operations Research **54**(1), 115–129 (2006)

[88] Hong, L.J., Nelson, B.L.: A brief introduction to optimization via simulation. In: M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, R.G. Ingalls (eds.) Proceedings of the 2009 Winter Simulation Conference (2009)

[89] Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. Journal of the Association for Computing Machinery **8**, 212–219 (1961)

[90] Hsu, J.C.: Multiple Comparisons: theory and methods. CRC Press, Boca Raton (1996)

[91] Hu, J., Fu, M.C., Marcus, S.I.: Stochastic optimization using model reference adaptive search. In: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, J.A. Joines (eds.) Proceedings of the 2005 Winter Simulation Conference, pp. 811–818 (2005)

[92] Hu, J., Fu, M.C., Marcus, S.I.: A model reference adaptive search method for global optimization. Operations Research **55**(3), 549–568 (2007)

[93] Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. Journal of Global Optimization **34**, 441–466 (2006)

[94] Humphrey, D.G., Wilson, J.R.: A revised simplex search procedure for stochastic simulation response-surface optimization. INFORMS Journal on Computing **12**(4), 272–283 (2000)

[95] Hunter, S.R., Pasupathy, R.: Optimal sampling laws for stochastically constrained simulation optimization on finite sets. INFORMS Journal on Computing **25**(3), 527–542 (2013)

[96] Hutchison, D.W., Hill, S.D.: Simulation optimization of airline delay with constraints. In: B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Rohrer (eds.) Proceedings of the 2001 Winter Simulation Conference, pp. 1017–1022 (2001)

[97] Huyer, W., Neumaier, A.: SNOBFIT – Stable noisy optimization by branch and fit. ACM Transactions on Mathematical Software **35**, 1–25 (2008)

[98] Irizarry, M.D.L.A., Wilson, J.R., Trevino, J.: A flexible simulation tool for manufacturing-cell design, II: Response surface analysis and case study. IIE Transactions (2001)

[99] Jacobson, S.H., Schruben, L.W.: Techniques for simulation response optimization. Operation Research Letters **8**, 1–9 (1989)

[100] Jia, Q.S., Ho, Y.C., Zhao, Q.C.: Comparison of selection rules for ordinal optimization. Mathematical and Computer Modelling **43**(9–10), 1150–1171 (2006)

[101] Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Application **79**, 157–181 (1993)

[102] Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**, 455–492 (1998)

[103] Jung, J.Y., Blau, G., Pekny, J.F., Reklaitis, G.V., Eversdyk, D.: A simulation based optimization approach to supply chain management under demand uncertainty. Computers & Chemical Engineering **28**, 2087–2106 (2004)

[104] Kabirian, A.: Continuous optimization via simulation using golden region search. Ph.D. thesis, Iowa State University (2009)

[105] Kabirian, A., Ólafsson, S.: Allocation of simulation runs for simulation optimization. In: S.G. Henderson, B. Biller, M.H. Hsieh, J. Shortle, J.D. Tew, R.R. Barton (eds.) Proceedings of the 2007 Winter Simulation Conference, pp. 363–371 (2007)

[106] Kabirian, A., Ólafsson, S.: Continuous optimization via simulation using golden region search. European Journal of Operational Research (2011)

[107] Kenne, J.P., Gharbi, A.: A simulation optimization approach in production planning of failure prone manufacturing systems. Journal of Intelligent Manufacturing **12**, 421–431 (2001)

[108] Khan, H.A., Zhang, Y., Ji, C., Stevens, C.J., Edwards, D.J., O'Brien, D.: Optimizing polyphase sequences for orthogonal netted radar. Signal Processing Letters, IEEE **13**(10), 589–592 (2006)

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

[110] Kim, S.H.: Comparison with a standard via fully sequential procedures. ACM Transactions on Modeling and Computer Simulation **15**(2), 155–174 (2005)

[111] Kim, S.H., Nelson, B.L.: Selecting the best system. In: S.G. Henderson, B.L. Nelson (eds.) Handbooks in Operations Research and Management Science: Simulation, chap. 17, pp. 501–534. Elsevier Science (2006)

[112] Kim, S.H., Nelson, B.L.: Recent advances in ranking and simulation. In: S.G. Henderson, B. Biller, M.H. Hsieh, J. Shortle, J.D. Tew, R.R. Barton (eds.) Proceedings of the 2007 Winter Simulation Conference, pp. 162–172 (2007)

[113] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)

[114] Kleijnen, J.P.C.: Simulation and optimization in production planning: A case study. Decision Support Systems **9**, 269–280 (1993)

[115] Kleijnen, J.P.C.: Design and Analysis of Simulation Experiments. Springer, New York (2008)

[116] Kleijnen, J.P.C.: Kriging metamodeling in simulation: A review. European Journal of Operational Research **192**(3), 707–716 (2009)

[117] Kleijnen, J.P.C., van Beers, W.C.M.: Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. European Journal of Operational Research **165**, 826–834 (2005)

[118] Kleijnen, J.P.C., Beers, W.C.M., van Nieuwenhuyse, I.: Expected improvement in efficient global optimization through bootstrapped kriging. Journal of Global Optimization **54**(1), 59–73 (2012)

[119] Kleinman, N.L., Hill, S.D., Ilenda, V.A.: Spsa/simmod optimization of air traffic delay cost. In: Proceedings of the 1997 American Control Conference, vol. 2, pp. 1121–1125 (1997)

[120] Köchel, P., Nieländer, U.: Simulation-based optimisation of multi-echelon inventory systems. International Journal of Production Economics **93–94**, 505–513 (2005)

[121] Kolda, T.G., Lewis, R.M., Torczon, V.J.: Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review **45**, 385–482 (2003)

[122] Kothandaraman, G., Rotea, M.A.: Simultaneous-perturbation-stochastic-approximation algorithm for parachute parameter estimation. Journal of Aircraft **42**(5), 1229–1235 (2005)

[123] Kroese, D.P., Hui, K.P., Nariai, S.: Network reliability optimization via the cross-entropy method. Reliability, IEEE Transactions on **56**(2), 275–287 (2007)

[124] Kroese, D.P., Porotsky, S., Rubinstein, R.Y.: The cross-entropy method for continuous multi-extremal optimization. Methodology and Computing in Applied Probability **8**(3), 383–407 (2006)

[125] Kulturel-Konak, S., Konak, A.: Simulation optimization embedded particle swarm optimization for reliable server assignment. In: B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, E. Yücesan (eds.) Proceedings of the 2010 Winter Simulation Conference, pp. 2897–2906 (2010)

[126] Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: A new tool for evolutionary computation. Kluwer Academic Publishers (2002)

[127] Lau, T.W.E., Ho, Y.C.: Universal alignment probabilities and subset selection for ordinal optimization. Journal of Optimization Theory and Applications **93**(3), 455–489 (1997)

[128] Law, A.M., Kelton, W.D.: Simulation modeling and analysis, 3rd edn. McGraw-Hill, Singapore (2000)

[129] Lee, L.H., Pujowidianto, N.A., Li, L.W., Chen, C.H., Yap, C.M.: Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints. IEEE Transactions on Automatic Control **57**(11), 2940–2945 (2012)

[130] Li, Y.: A simulation-based evolutionary approach to LNA circuit design optimization. Applied Mathematics and Computation **209**(1), 57–67 (2009)

[131] Lucidi, S., Sciandrone, M.: On the global convergence of derivative-free methods for unconstrained minimization. SIAM Journal on Optimization **13**, 97–116 (2002)

[132] Lutz, C.M., Davis, K.R., Sun, M.: Determining buffer location and size in production lines using tabu search. European Journal of Operational Research **106**, 301–316 (1998)

[133] Martí, R., Laguna, M., Glover, F.: Principles of scatter search. European Journal of Operational Research **169**(2), 359–372 (2006)

[134] Maryak, J.L., Chin, D.C.: Global random optimization by simulataneous perturbation stochastic approximation. IEEE Transactions on Automatic Control **53**, 780–783 (2008)

[135] Meketon, M.S.: Optimization in simulation: A survey of recent results. In: A. Thesen, H. Grant, W.D. Kelton (eds.) Proceedings of the 1987 Winter Simulation Conference, pp. 58–67 (1987)

[136] Merhof, D., Soza, G., Stadlbauer, A., Greiner, G., Nimsky, C.: Correction of susceptibility artifacts in diffusion tensor data using nonlinear registration. Medical Image Analysis **11**(6), 588–603 (2007)

[137] Merton, R.C.: On the pricing of corporate debt: The risk structure of interest rates. The Journal of Finance **29**(2), 449–470 (1974)

[138] Mishra, V., Bhatnagar, S., Hemachandra, N.: Discrete parameter simulation optimization algorithms with applications to admission control with dependent service times. In: Proceedings of the 46th IEEE Conference on Decision and Control, pp. 2986–2991. New Orleans, LA (2007)

[139] Mockus, J.: Bayesian Approach to Global Optimization. Kluwer Academic Publishers (1989)

[140] Mockus, J., Tiesis, V., Zilinskas, A.: Towards Global Optimisation, vol. 2, chap. The application of Bayesian Methods for Seeking the Extremum, pp. 117–128. North-Holland, Amsterdam (1978)

[141] Moré, J., Wild, S.: Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization **20**, 172–191 (2009)

[142] Myers, R.H., Montgomery, D.C., Anderson-Cook, C.M.: Response Surface Methodology: Process and Product Optimization Using Designed Experiments. Wiley Series in Probability and Statistics. John Wiley & Sons (2009)

[143] Neddermeijer, H.G., Oortmarssen, G.J.V., Piersma, N., Dekker, R.: A framework for response surface methodology for simulation optimization. In: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (eds.) Proceedings of the 2000 Winter Simulation Conference, pp. 129–136 (2000)

[144] Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer Journal **7**, 308–313 (1965)

[145] Nelson, B.L.: Optimization via simulation over discrete decision variables. Tutorials in operations research **7**, 193–207 (2010)

[146] Nelson, B.L., Goldsman, D.: Comparisons with a standard in simulation experiments. Management Science **47**(3), 449–463 (2001)

[147] Nicolai, R., Dekker, R.: Automated response surface methodology for simulation optimization models with unknown variance. Quality Technology and Quantitative Management **6**(3), 325–352 (2009)

[148] Ö. Yalçinkaya, G.M.B.: Modelling and optimization of average travel time for a metro line by simulation and response surface methodology. European Journal of Operational Research **196**, 225–233 (2009)

[149] Ólafsson, S.: Metaheuristics. In: S. Henderson, B. Nelson (eds.) Handbook in Operations Research and Management Science: Simulation 13, pp. 633–654. Elsevier (2006)

[150] http://www.orms-today.org/surveys/Simulation/Simulation.html

[151] Osorio, C., Bierlaire, M.: A simulation-based optimization approach to perform urban traffic control. In: Proceedings of the Triennial Symposium on Transportation Analysis (2010)

[152] Pasupathy, R., Ghosh, S.: Simulation optimization: A concise overview and implementation guide. Tutorials in operations research **10**, 122–150 (2013)

[153] Pasupathy, R., Henderson, S.G.: SIMOPT: A library of simulation-optimization problems. In: S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, e. M. Fu (eds.) Proceedings of the 2011 Winter Simulation Conference (2011)

[154] Pasupathy, R., Kim, S.: The stochastic root finding problem: Overview, solutions, and open questions. ACM Transactions on Modeling and Computer Simulation **21**(3), 19:1–19:23 (2011)

[155] Peters, J., Vijayakumar, S., Schaal, S.: Reinforcement learning for humanoid robotics. In: Third IEEE-RAS International Conference on Humanoid Robots, pp. 1–20. Karlsruhe, Germany (2003)

[156] Pflug, G.C.: Optimization of stochastic models: The interface between simulation and optimization. Kluwer Academic Publishers (1996)

[157] Powell, W.B.: http://www.castlelab.princeton.edu/cso.htm (accessed October 23, 2013)

[158] Powell, W.B., Ryzhov, I.O.: Optimal Learning. John Wiley & Sons (2012)

[159] Prakash, P., Deng, G., Converse, M.C., Webster, J.G., Mahvi, D.M., Ferris, M.C.: Design optimization of a robust sleeve antenna for hepatic microwave ablation. Physics in Medicine and Biology **53**, 1057–1069 (2008)

[160] Radac, M.B., Precup, R.E., Petriu, E.M., Preitl, S.: Application of ift and spsa to servo system control. IEEE Transactions on Neural Networks **22**(12), 2363–2375 (2011)

[161] Rall, L.B.: Automatic Differentiation: Techniques and Applications, *Lecture Notes in Computer Science*, vol. 120. Springer, Berlin (1981). DOI 10.1007/3-540-10861-0

[162] Ramanathan, S.P., Mukherjee, S., Dahule, R.K., Ghosh, S., Rahman, I., Tambe, S.S., Ravetkar, D.D., Kulkarni, B.D.: Optimization of continuous distillation columns using stochastic optimization approaches. Transactions of the Institution of Chemical Engineers **79**, 310–322 (2001)

[163] Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press (2006)

[164] Reeves, C.R.: Genetic algorithms for the operations researcher. INFORMS Journal on Computing **9**(3), 231–250 (1997)

[165] Renotte, C., Vande Wouwer, A.: Stochastic approximation techniques applied to parameter estimation in a biological model. In: Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings of the Second IEEE International Workshop on, pp. 261–265. IEEE (2003)

[166] Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization **56**, 1247–1293 (2013)

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

[168] Romero, P.A., Krause, A., Arnold, F.H.: Navigating the protein fitness landscape with gaussian processes. Proceedings of the National Academy of Sciences (PNAS) **110**(3) (2013). DOI 10.1073/pnas.1215251110

[169] Roustant, O., Ginsbourger, D., Deville, Y.: DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. Journal of Statistical Software **51**(1), 1–55 (2012)

[170] Rubinstein, R.: The cross-entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability **1**, 127–190 (1999)

[171] Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning. Springer-Verlag, New York (2004)

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

[173] Sacks, J., Schiller, S.B., Welch, W.J.: Designs for computer experiments. Technometrics **31**, 41–47 (1989)

[174] Safizadeh, M.H.: Optimization in simulation: Current issues and the future outlook. Naval Research Logistics **37**, 807–825 (1990)

[175] Sahinidis, N.V.: Optimization under uncertainty: State-of-the-art and opportunities. Computers & Chemical Engineering **28**(6–7), 971–983 (2004)

[176] Schwartz, J.D., Wang, W., Rivera, D.E.: Simulation-based optimization of process control policies for inventory management in supply chains. Automatica **42**, 1311–1320 (2006)

[177] Scott, W., Frazier, P.I., Powell, W.: The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. SIAM Journal on Optimization **21**(3), 996–1026 (2011)

[178] Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin–Madison (2010)

[179] Shapiro, A.: Asymptotic analysis of stochastic programs. Annals of Operations Research **30**, 169–186 (1991)

[180] Shapiro, A.: Simulation based optimization. In: J.M. Charnes, D.J. Morrice, D.T. Brunner, J.J. Swain (eds.) Proceedings of the 1996 Winter Simulation Conference, pp. 332–336 (1996)

[181] Shi, L., Ólafsson: Nested Partitions Optimization: Methodology and Applications, *International Series in Operations Research & Management Science*, vol. 109. Springer (2007)

[182] Shi, L., Ólafsson, S.: Nested partitions method for stochastic optimization. Methodology and Computing in Applied Probability **2**, 271–291 (2000)

[183] Song, Y., Grizzle, J.W.: The extended kalman filter as a local asymptotic observer for discrete-time nonlinear systems. Journal

of Mathematical Systems, Estimation, and Control **5**(1), 59–78 (1995)

[184] Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control **37**, 332–341 (1992)

[185] Spall, J.C.: Chapter 7: Simultaneous Perturbation Stochastic Approximation. In: Introduction to stochastic search and optimization: Estimation, simulation, and control. Wiley-Interscience (2003)

[186] Spall, J.C.: Introduction to stochastic search and optimization: Estimation, simulation, and control. Wiley-Interscience (2003)

[187] Spall, J.C.: Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm. IEEE Transactions on Automatic Control **54**(6), 1216–1229 (2009)

[188] Spall, J.C.: Stochastic optimization. In: J.E. Gentle, W.K. Härdle, Y. Mori (eds.) Handbook of Computational Statistics: Concepts and Methods, second edn., chap. 7, pp. 173–201. Springer (2012)

[189] Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.: Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory **58**(5), 3250–3265 (2012)

[190] Stephens, C.P., Baritompa, W.: Global optimization requires global information. Journal of Optimization Theory and Applications **96**, 575–588 (1998)

[191] Swisher, J.R., Hyden, P.D., Jacobson, S.H., Schruben, L.W.: A survey of simulation optimization techniques and procedures. In: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (eds.) Proceedings of the 2000 Winter Simulation Conference (2000)

[192] Syberfeldt, A., Lidberg, S.: Real-world simulation-based manufacturing optimization using cuckoo search. In: C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, A. Uhrmacher (eds.) Proceedings of the 2012 Winter Simulation Conference (2012)

[193] Tein, L.H., Ramli, R.: Recent advancements of nurse scheduling models and a potential path. In: Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and its Applications, pp. 395–409 (2010)

[194] Tekin, E., Sabuncuoglu, I.: Simulation optimization: A comprehensive review on theory and applications. IIE Transactions **36**, 1067–1081 (2004)

[195] Teng, S., Lee, L.H., Chew, E.P.: Multi-objective ordinal optimization for simulation optimization problems. Automatica **43**(11), 1884–1895 (2007)

[196] Trosset, M.W.: On the use of direct search methods for stochastic optimization. Tech. rep., Rice University, Houston, TX (2000)

[197] Vande Wouwer, A., Renotte, ., Bogaerts, P., Remy, M.: Application of spsa techniques in nonlinear system identification. In: Proceedings of the European Control Conference, p. 2835 (2001)

[198] Wang, H., Pasupathy, R., Schmeiser, B.W.: Integer-ordered simulation optimization using R-SPLINE: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. ACM Transactions on Modeling and Computer Simulation **23**, 17:1–17:24 (2012)

[199] Wang, Q., Spall, J.C.: Discrete simultaneous perturbation stochastic approximation on loss functions with noisy measurements. In: Proceedings of the American Control Conference, pp. 4520–4525. IEEE, San Francisco (2011)

[200] Whitley, D.: A genetic algorithm tutorial. Statistics and Computing **4**, 65–85 (1994)

[201] Xie, J., Frazier, P.I.: Sequential bayes-optimal policies for multiple comparisons with a known standard. Operations Research **61**(5), 1174–1189 (2013)

[202] Xie, J., Frazier, P.I., Sankaran, S., Marsden, A., Elmohamed, S.: Optimization of computationally expensive simulations with gaussian processes and parameter uncertainty: Application to cardiovascular surgery. In: 50th Annual Allerton Conference on Communication, Control, and Computing (2012)

[203] Xing, X.Q., Damodaran, M.: Assessment of simultaneous perturbation stochastic approximation method for wing design optimization. Journal of Aircraft **39**, 379–381 (2002)

[204] Xing, X.Q., Damodaran, M.: Application of simultaneous perturbation stochastic approximation method for aerodynamic shape design optimization. AIAA Journal **43**(2), 284–294 (2005)

[205] Xing, X.Q., Damodaran, M.: Inverse design of transonic airfoils using parallel simultaneous perturbation stochastic approximation. Journal of Aircraft **42**(2), 568–570 (2005)

[206] Xu, J., Nelson, B.L., Hong, L.J.: Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. ACM Transactions on Modeling and Computer Simulation **20**(1), 1–29 (2010)

[207] Xu, J., Nelson, B.L., Hong, L.J.: An adaptive hypberbox algorithm for high-dimensional discrete optimization via simulation problems. INFORMS Journal on Computing **25**(1), 133–146 (2013)

[208] Yan, D., Mukai, H.: Stochastic discrete optimization. SIAM Journal on Control and Optimization **30**, 594–612 (1992)

[209] Yang, X.S., Deb, S.: Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation **1**(4), 330–343 (2010)

[210] Yeomans, J.S.: Solid waste planning under uncertainty using evolutionary simulation-optimization. Socio-Economic Planning Sciences **41**, 38–60 (2007)

[211] Yun, I., Park, B.: Application of stochastic optimization method for an urban corridor. In: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto (eds.) Proceedings of the 2010 Winter Simulation Conference, pp. 1493–1499 (2010)