A Framework for Locally Convergent Random-Search Algorithms for Discrete Optimization via Simulation

L. JEFF HONG
The Hong Kong University of Science and Technology
and
BARRY L. NELSON
Northwestern University

The goal of this article is to provide a general framework for locally convergent random-search algorithms for stochastic optimization problems when the objective function is embedded in a stochastic simulation and the decision variables are integer ordered. The framework guarantees desirable asymptotic properties, including almost-sure convergence and known rate of convergence, for any algorithms that conform to its mild conditions. Within this framework, algorithm designers can incorporate sophisticated search schemes and complicated statistical procedures to design new algorithms.

Categories and Subject Descriptors: G.1.6 [**Numerical Analysis**]: Optimization; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Random search, discrete stochastic optimization

ACM Reference Format:

Hong, L. J. and Nelson, B. L. 2007. A framework for locally convergent random-search algorithms for discrete optimization via simulation. ACM Trans. Model. Comput. Simul. 17, 4, Article 19 (September 2007), 22 pages. DOI = 10.1145/1276927.1276932 http://doi.acm.org/10.1145/1276927.1276932

This research was supported in part by Hong Kong Research Grants Council grant Nos. DAG04/05.EG04 and CERG 613305, National Science Foundation Grant DMI-0217690 and General Motors R & D.

Author's addresses: L. J. Hong, Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China; email: hongl@ust.hk; B. L. Nelson, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208-3119; email: nelsonb@northwestern.edu.

1. INTRODUCTION

Optimization via simulation (OvS) refers to optimization problems where the objective function value is the expected value of performance from a stochastic computer simulation. OvS applications have been increasing in areas such as manufacturing (e.g., Vogt [2004]), supply-chain management (e.g., Truong and Azadivar [2003]), logistics (e.g., Wieland and Holden [2003]), telecommunications (e.g., Baras [2003]), and project management (e.g., April et al. [2004]).

The burst of OvS applications creates a demand for efficient OvS algorithms. Many algorithms have been designed in the past ten years (see Fu [2002] for a thorough review). Typically, algorithms reported in the simulation literature are simple with nice mathematical properties, including guaranteed convergence and known rate of convergence, but they are often not sophisticated enough to solve some practical problems. Algorithms used in commercial software work well for many practical problems, but typically provide no provable convergence and can be misled by noise in the simulation experiment. One reason for the gap between academic research and software practice is the nature of OvS problems. To solve OvS problems correctly and efficiently, one needs to consider both the stochastic side and the algorithmic side of the problem, but few researchers are experts in both fields. The algorithms in the simulation literature are often developed by experts in statistics and stochastic processes, while those used in commercial software are typically developed by experts in (deterministic) optimization and computer science (see, e.g., April et al. [2004]).

To bridge the gap, the most important task, from our point of view, is to create frameworks for different types of OvS problems; frameworks that provide a set of conditions so that all algorithms designed according to a particular framework are guaranteed to have certain asymptotic properties, such as, convergence to an optimal solution at a known rate. The conditions should be mild enough that they impose few constraints on the design of algorithms. This will allow researchers in optimization and computer science to design efficient algorithms that conform to one of the frameworks and thereby guarantee to also have the desired statistical properties.

At this point, it is worthwhile to discuss the importance of convergence properties. Such properties guarantee an algorithm converges to the desired solutions at an acceptable rate when the number of simulation observations goes to infinity. In practice, however, all algorithms stop in a finite amount of time, so why are convergence properties desirable? The following statement from Andradóttir [2002] answers this question.

> Although this [convergence] performance guarantee does not assure that the algorithm will return a "good" estimated optimal solution (because additional computational effort may be required), it is certainly a reassuring property to have. From a different perspective, it is worrisome to use a simulation optimization algorithm in practice that is not known to converge even if an infinite amount of computational effort is expended!

Therefore, it is important to design useful frameworks and to develop efficient algorithms within them. The resulting algorithms will have not only good finite-time performance, but also "reassuring" asymptotic properties.

In this article, we propose such a framework to solve OvS problems with integer-ordered decision variables, also called discrete OvS or DOvS problems. DOvS problems are very common in practice where decisions are related to items that come in discrete units, for example, products, machines, or vehicles. Our framework guarantees that algorithms that conform to it converge with probability 1 to the set of local optimal solutions for DOvS problems with either a finite or countably infinite number of feasible solutions.

Optimization with integer-ordered decision variables is also studied in the integer programming (IP) literature (see, e.g., Wolsey [1998]). A typical IP problem has a closed-form objective function that can be evaluated even when integrality constraints on the decision variables are violated. Therefore, algorithms that depend on the relaxation of integrality constraints, such as branch-and-bound, are often used to solve IP problems. The objective function of a DOvS problem, however, is embedded in a simulation model and typically cannot be evaluated at fractional values of the decision variables. For instance, it is typically not possible to evaluate a manufacturing station with 3.5 machines. Therefore, algorithms that depend on the relaxation of integrality constraints cannot be applied directly. Moreover, the objective functions of DOvS problems can only be estimated with noise when the output from the simulation is stochastic. Therefore, the allocation of simulation effort to each function evaluation also needs to be considered. These properties make DOvS problems drastically different from typical IP problems.

Many methods have been proposed to solve DOvS problems. When the number of feasible solutions is small, we can simulate all feasible ones and select the best. Several approaches have been proposed in the literature: mutiple comparisons (see, e.g., Bechhofer et al. [1995]), ranking and selection (R&S), (see, e.g., Nelson et al. [2001] and Chick and Inoue [2001]), and optimal computational budget allocation (OCBA) (see, e.g., Chen et al. [2000]). When the number of feasible solutions becomes large, there are a number of approaches to solving the problem, including sample-path approximation [Kleywegt et al. 2001] and random-search algorithms (see, e.g., the stochastic ruler method of Yan and Mukai [1992], the simulated annealing method of Alrefaei and Andradóttir [1999], the stochastic comparison method of Gong et al. [1999], the nested partitions method of Shi and Ólafsson [2000], and the COMPASS algorithm of Hong and Nelson [2006]).

Among random-search algorithms, the stochastic ruler, simulated annealing, stochastic comparison, and nested partitions methods are all globally convergent; that is, they can be shown to converge to the set of global optimal solutions as the simulation effort goes to infinity. However, global convergence is typically achieved by allocating an infinite number of simulation observations to *all feasible solutions* in the limit (an exception is Andradóttir [2006]). COMPASS, on the other hand, is locally convergent. On each iteration COMPASS samples a collection of new solutions and allocates simulation observations to all sampled solutions using a simulation-allocation rule. The convergence is achieved by showing that *all sampled solutions* obtain an infinite number of simulation observations in the limit. Since only a small fraction of feasible solutions is typically sampled, COMPASS is efficient in finding *local* optimal solutions.

Furthermore, most globally convergent algorithms in the literature can only solve problems with a finite number of feasible solutions, but COMPASS can solve those with both a finite or countably infinite number.

For globally convergent random-search algorithms, Andradóttir [1999] provides a unified scheme to estimate the optimal objective function value and shows that the resulting algorithms satisfy a central limit theorem. This scheme may be viewed as a framework for globally convergent random-search algorithms for DOvS problems. In this article we develop a framework for solving DOvS problems using locally convergent random-search (LCRS) algorithms. The framework ensures that the algorithms that conform to it have desirable asymptotic properties. Moreover, it separates the algorithmic and statistical sides of the algorithm design and allows experts in both fields to contribute.

Though our work focuses on asymptotic guarantees, it is worthwhile to notice that there also exist other finite-time statistical guarantees in the OvS literature. Boesel et al. [2003] implement an R&S procedure when the optimization process stops, to ensure that the reported optimal solution is at least the best among all sampled solutions with certain statistical confidence. Hong and Nelson [2007] imbed R&S procedures in the optimization process to achieve the same goal whenever the optimization stops. The OCBA procedure of Chen et al. [2000] can also be used for the same purposes. These statistical guarantees typically do not ensure optimality of solution as do asymptotic guarantees. They can be easily incorporated in our frameworks to improve the algorithms' finite-time performances.

The remainder of the article is organized as follows: In Section 2 we present a framework for LCRS algorithms. The local convergence and asymptotic normality of the framework are established in Sections 3 and 4. In Section 5, we discuss a specific algorithm design issue for problems with a countably infinite number of feasible solutions. As an application of the framework, we revise the COMPASS algorithm of Hong and Nelson [2006] and provide some numerical results to compare the two algorithms in Section 6.

2. A FRAMEWORK FOR LCRS ALGORITHMS

Consider the DOvS problem

$$\min_{\mathbf{x}\in\Theta} \ \mathrm{E}_{\psi}\left[G\left(\mathbf{x},\psi\right)\right], \tag{1}$$

where $\Theta \subset \mathcal{Z}^d$ and $\mathcal{Z}^d$ is the set of d-dimensional vectors with integer elements. To avoid triviality, we assume that $|\Theta| > 0$, where $|\cdot|$ denotes the cardinality of a set. The quantity ψ represents the stochastic input to the simulation, and its distribution may depend on $\mathbf{x}$. We assume that $G(\mathbf{x},\psi)$ is measurable and integrable with respect to the distribution of ψ for all $\mathbf{x}\in\Theta$. Furthermore, we let $g(\mathbf{x}) = \mathrm{E}_{\psi}\left[G\left(\mathbf{x},\psi\right)\right]$ and assume that $g(\mathbf{x})$ cannot be evaluated easily (or at all), but that the random variable $G(\mathbf{x},\psi)$ can be observed via a simulation experiment at $\mathbf{x}$. The ith observation of $G(\mathbf{x},\psi)$ is denoted by $G_i(\mathbf{x})$.

To design locally convergent algorithms, we first need to define local optimality. We adopt the definition of Hong and Nelson [2006]. Let

$$\mathcal{L}N(\mathbf{x}) = \left\{\mathbf{y} : \mathbf{y}\in\Theta \text{ and } \|\mathbf{x}-\mathbf{y}\| \le 1\right\}$$

be the *local neighborhood* of $\mathbf{x} \in \Theta$, where $\|\mathbf{x}-\mathbf{y}\|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. Notice that $|\mathcal{LN}(\mathbf{x})| \leq 2d+1$. We define $\mathbf{x}$ as a *local minimizer* if $\mathbf{x} \in \Theta$ and $g(\mathbf{x}) \leq g(\mathbf{y})$ for all $\mathbf{y} \in \mathcal{L}N(\mathbf{x})$. Let $\mathcal{M}$ denote the set of local minimizers of the function g in Θ; locally convergent algorithms guarantee to converge to solutions in $\mathcal{M}$.

Throughout this work, we use "sampling" to mean choosing a feasible solution $\mathbf{x}$ (typically randomly), and use "estimation" to mean generating simulation observations $G(\mathbf{x})$ to evaluate the value of a solution. Any random-search algorithm for DOvS problems has two critical elements: a sampling scheme and an estimation scheme. On each iteration, the sampling scheme determines what solutions to sample and the estimation scheme determines what solutions to simulate and how many observations to allocate to each solution. Specially, on iteration k, the sampling scheme determines a neighborhood $\mathcal{C}_k$ from which the solutions are sampled, a sampling distribution F_k which is defined on $\mathcal{C}_k$ and may depend on all past information, and m_k, which is the number of solutions to sample. Typically, the solutions are sampled independently of both previous iterations and of each other, so it is possible that the sampled solutions may include duplicates. On iteration k, the estimation scheme determines an estimation set $\mathcal{E}_k$ which includes all solutions that could be simulated on iteration k, and $a_k(\mathbf{x})$, which is the additional number of simulation observations allocated to $\mathbf{x}$ on iteration k for all $\mathbf{x} \in \mathcal{E}_k$. Notice that it is possible that $a_k(\mathbf{x}) = 0$ if $\mathbf{x}$ already has enough observations.

Let $a_k(\mathbf{x}) = 0$ if $\mathbf{x} \notin \mathcal{E}_k$, and let $N_k(\mathbf{x}) = \sum_{i=0}^{k} a_i(\mathbf{x})$ for all $\mathbf{x} \in \Theta$. Then $N_k(\mathbf{x})$ is the total number of simulation observations allocated to $\mathbf{x}$ through iteration k. We also let $\bar{G}_k(\mathbf{x}) = \sum_{i=1}^{N_k(\mathbf{x})} G_i(\mathbf{x})/N_k(\mathbf{x})$ if $N_k(\mathbf{x}) > 0$ and let $\bar{G}_k(\mathbf{x}) = 0$ if $N_k(\mathbf{x}) = 0$ for all $k = 1, 2, \ldots$. Then the following is our framework for LCRS algorithms.

The Generic Algorithm

Step 0. Let $\mathbf{x}_0 \in \Theta$ be the starting solution provided by the user. Set the iteration count $k = 0$. Let $N_0(\mathbf{x}) = 0$ and $\bar{G}_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Theta$, and let $\mathcal{E}_0 = \emptyset$. Moreover, let the current sample-best solution be $\widehat{\mathbf{x}}_0^* = \mathbf{x}_0$, let the set of sampled solutions be $\mathcal{S}_0 = \{\mathbf{x}_0\}$, and let $\mathcal{S}(0) = \mathcal{S}_0$.

Step 1. Let $k = k + 1$. Determine $\mathcal{C}_k$ and F_k according to the sampling scheme. Independently of all previous iterations, sample a set of m_k solutions from $\mathcal{C}_k$ using the sampling distribution F_k. Let $\mathcal{S}_k$ be the set of unique solutions (i.e., remove the duplicate solutions) from the m_k newly sampled solutions, and let $\mathcal{S}(k) = \mathcal{S}(k-1) \cup \mathcal{S}_k$.

Step 2. Determine $\mathcal{E}_k \subset \mathcal{S}(k)$ and $a_k(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{E}_k$ according to the estimation scheme. For all $\mathbf{x} \in \mathcal{E}_k$, take $a_k(\mathbf{x})$ simulation observations and update $N_k(\mathbf{x})$ and $\bar{G}_k(\mathbf{x})$. For all $\mathbf{x} \notin \mathcal{E}_k$, let $N_k(\mathbf{x}) = N_{k-1}(\mathbf{x})$ and $\bar{G}_k(\mathbf{x}) = \bar{G}_{k-1}(\mathbf{x})$.

Step 3. Let $\widehat{\mathbf{x}}_k^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{E}_k} \bar{G}_k(\mathbf{x})$. If there is more than one solution having the smallest sample mean $\min_{\mathbf{x}\in\mathcal{E}_k} \bar{G}_k(\mathbf{x})$, then any such solution can be chosen as $\widehat{\mathbf{x}}_k^*$. Go to Step 1.

Remark. Step 1 permits the solutions in $\mathcal{S}_k$ to have been sampled in previous iterations.

Note that $\mathcal{S}(k) = \bigcup_{i=0}^{k} \mathcal{S}_i$. This is the set of solutions sampled through iteration k. Let $\mathcal{E}(k) = \bigcup_{i=0}^{k} \mathcal{E}_i$ be the set of solutions estimated through iteration

k. Let $\mathcal{S}(\infty) = \bigcup_{i=0}^{\infty} \mathcal{S}_i$ and $\mathcal{E}(\infty) = \bigcup_{i=0}^{\infty} \mathcal{E}_i$. We require the algorithm to satisfy the following conditions.

Condition 1. The sampling scheme satisfies the following requirements:

(1) There exists a deterministic sequence of finite sets, denoted by $\mathcal{B}_1, \mathcal{B}_2, \ldots$, such that $\mathcal{C}_k \subset \mathcal{B}_k$ for all $k = 1, 2, \ldots$, $\mathcal{B}_1 \subset \mathcal{B}_2 \subset \cdots$, and $\Theta \subset \bigcup_{k=1}^{\infty} \mathcal{B}_k$; and

(2) the sampling distribution F_k guarantees that $\Pr\{\mathbf{x} \in \mathcal{S}_k\} \geq \epsilon$ for all $\mathbf{x} \in [\mathcal{LN}(\widehat{\mathbf{x}}_{k-1}^*) \setminus \mathcal{E}(k-1)]$ for some $\epsilon > 0$ that is independent of k.

Condition 1 is on the sampling scheme. The first requirement ensures that $\mathcal{C}_k$ can be bounded by a sequence of sets $\mathcal{B}_k$. The sets $\mathcal{B}_k$, $k = 1, 2, \ldots$, are nested and deterministic. Moreover, $\mathcal{B}_k$ is finite for each k, but $\bigcup_{k=1}^{\infty} \mathcal{B}_k$ may be an infinite set which contains all feasible solutions. When Θ is a finite set, then we may let $\mathcal{B}_k = \Theta$ for all $k = 1, 2, \ldots$ and the requirement is easily satisfied.

When Θ is countably infinite, this requirement prevents the sampling scheme from sampling points that are arbitrarily far away. We must design $\mathcal{C}_k$ carefully to satisfy this requirement. For instance, Hong and Nelson [2006] start with $\mathcal{C}_0$ in a predetermined hyperbox that includes $\mathbf{x}_0$. Then on each iteration k, if $\hat{\mathbf{x}}_k^*$ is within a certain distance from a boundary of the hyperbox, then the boundary will be extended by a value of Δ in the next iteration. Suppose that the initial hyperbox is $\prod_{j=1}^{d}[\ell_j, u_j]$. Then we may let $\mathcal{B}_k = \mathcal{Z}^d \cap \prod_{j=1}^{d}[\ell_j - k\Delta, u_j + k\Delta]$ and $\mathcal{C}_k \subset \mathcal{B}_k$ for all $k = 1, 2, \ldots$. Notice that $|\mathcal{B}_k| \leq (c + 2k\Delta)^d$ for some $c > 0$.

Another approach to satisfying the first requirement is to mandate that any solution $\mathbf{x} \in \mathcal{C}_k$ satisfies $\|\mathbf{x} - \widehat{\mathbf{x}}_{k-1}^*\| \leq \Delta$ for some $\Delta > 0$, which means that only those solutions that are within Δ distance from the current sample-best solution can be sampled. Then we may let $\mathcal{B}_k = \{\mathbf{x} \in \mathcal{Z}^d : \|\mathbf{x} - \mathbf{x}_0\| \leq k\Delta\}$ and $|\mathcal{B}_k| \leq (1 + 2k\Delta)^d$.

The second requirement of Condition 1 ensures that all local neighbors of $\widehat{\mathbf{x}}_{k-1}^*$ that have not been estimated through iteration $k-1$ are included in the sampling neighborhood $\mathcal{C}_k$ and have a probability of being chosen that is bounded away from zero. Therefore, if $\widehat{\mathbf{x}}_k^*$ equals $\mathbf{x}$ infinitely often (i.o.), all local neighbors of $\mathbf{x}$ will be estimated in the limit with probability 1 (w.p.1). A more precise statement of this property is given in Lemma 3.1 to follow.

Condition 2. The estimation scheme satisfies the following requirements:

(1) $\mathcal{E}_k$ is a subset of $\mathcal{S}(k)$;

(2) $\mathcal{E}_k$ contains $\mathbf{x}_0$, $\mathcal{LN}(\widehat{\mathbf{x}}_{k-1}^*) \cap \mathcal{E}(k-1)$ and $\mathcal{S}_k$;

(3) $a_k(\mathbf{x})$ is allocated such that $\min_{\mathbf{x}\in\mathcal{E}_k} N_k(\mathbf{x}) \geq 1$ for all $k = 1, 2, \ldots$ and $\min_{\mathbf{x}\in\mathcal{E}_k} N_k(\mathbf{x}) \to \infty$ w.p.1 as $k \to \infty$; and

(4) $|\mathcal{E}(\infty)| < \infty$ w.p.1.

Condition 2 is on the estimation scheme. The first two requirements are that $\mathcal{E}_k$ contains only solutions that have been sampled through iteration k, and that it contains at least $\mathbf{x}_0$, that is, all local neighbors of $\widehat{\mathbf{x}}_{k-1}^*$ that have been estimated through iteration $k-1$ (including $\widehat{\mathbf{x}}_{k-1}^*$), and all newly sampled solutions.

The third requirement of Condition 2 is that all solutions in $\mathcal{E}_k$ have at least one simulation observation so that the solution can be estimated, and the numbers of simulation observations allocated to solutions in $\mathcal{E}_k$ go to infinity, with probability 1 as $k \to \infty$ so that the solutions can be estimated without noise in the limit.

The fourth requirement of Condition 2 is that only a finite number of solutions are estimated in the limit. If $|\Theta| < \infty$, then this requirement is satisfied since $\mathcal{E}(\infty) \subset \Theta$. If Θ is countably infinite, then the algorithm needs to be designed more carefully to satisfy the condition. In Section 5, we present a design approach ensuring that $|\mathcal{E}(\infty)| < \infty$ w.p.1 when Θ is countably infinite.

In the generic algorithm, Condition 1 is on the sampling scheme, and Condition 2 on the estimation scheme. They are both mild conditions, and leave plenty of room for algorithm designers to produce efficient algorithms. Although the OvS literature focuses on general-purpose optimization algorithms which require few assumptions on the structure of the problem, our framework may also be used to design special-purpose algorithms that solve a particular class of DOvS problems.

The generic algorithm intentionally separates the sampling scheme and estimation scheme. The sampling scheme is more related to algorithm design, while the estimation scheme is more related to statistical design. This separation allows experts from different fields to cooperate to develop efficient algorithms. For instance, experts in deterministic optimization may use sophisticated methods, such as, neural networks or efficient metaheuristics, to guide the sampling process, while experts in statistics may design procedures based on techniques like R&S or OCBA to estimate solutions. Through the framework they can be combined to generate efficient DOvS algorithms.

Both Conditions 1 and 2 involve $\mathcal{E}(k)$, which is the set of solutions estimated through iteration k. Notice that the use of $\mathcal{E}(k)$ in Condition 1 is very limited: It only requires that local neighbors not yet sampled have a positive probability of being sampled. Many algorithms allow all local neighbors to be sampled, so clearly satisfy this condition. But the role of $\mathcal{E}(k)$ in Condition 2 is critical, and requires the algorithm to retain information on all solutions that have been estimated, which may become a very large set when the solution space is large. While this requirement would seem unreasonable for a deterministic random-search algorithm, it is reasonable for the design of convergent DOvS algorithms because the simulation outputs are noisy. If data is not retained, then to obtain convergence more and more observations must be acquired from solutions each time they are sampled to overcome the noise. Since simulation experiments are often more time consuming than retaining and searching past information, it makes sense to accumulate past observations (but it is important that they be stored and searched efficiently).

The framework can be used not only to design new algorithms, but also to simplify existing ones. We may check the conditions of an LCRS algorithm to see which are unnecessary compared to Conditions 1 and 2, and think about ways to remove them. One example is the COMPASS algorithm of Hong and Nelson [2006] which estimates all solutions that have ever been sampled on each iteration. According to Condition 2, however, only a portion of the sampled solutions need be estimated to guarantee convergence. Therefore, we may reduce the number of solutions that are estimated on each iteration to make COMPASS more efficient. This revision of the COMPASS algorithm will be discussed in Section 6.

3. LOCAL CONVERGENCE OF THE GENERIC ALGORITHM

To establish local convergence of the generic algorithm, we make following assumptions on $g(\mathbf{x})$ and $G_i(\mathbf{x})$:

Assumption 1. There exists a positive constant δ_0 such that the level set $\mathcal{L} = \{\mathbf{x} \in \Theta : g(\mathbf{x}) \leq g(\mathbf{x}_0) + \delta_0\}$ is finite.

Notice that $\mathcal{L} \subset \Theta$. When Θ is finite, Assumption 1 always holds. When Θ is countably infinite, Assumption 1 requires that there exists a good starting solution such that there are only finitely many better ones. Assumption 1 often holds, since a simulation study has a benchmark system configuration which may be the current one, and solutions that are far from the benchmark are typically inferior to the benchmark. Though the assumption often holds, verifying it when Θ is countably infinite is difficult.

Assumption 2. For every $\mathbf{x} \in \Theta$,

$$\lim_{r\to\infty} \frac{1}{r} \sum_{i=1}^{r} G_i(\mathbf{x}) = g(\mathbf{x}) \quad \text{w.p.1.}$$

Assumption 2 implies that the sample mean of $G(\mathbf{x}, \psi)$ is an appropriate estimator of $g(\mathbf{x})$. In the case of terminating simulation, $G_i(\mathbf{x})$, $i = 1, 2, \ldots,$ are independent and identically distributed and Assumption 2 is satisfied according to the strong law of large numbers [Durrett 1995]. In the case of steady-state simulation, $G_i(\mathbf{x})$, $i = 1, 2, \ldots,$ are observations from a long simulation run of a process with a limiting distribution and $g(\mathbf{x})$ is the steady-state mean response of the system at $\mathbf{x}$. Under certain conditions, Assumption 2 is satisfied in this setting as well (e.g., Law and Kelton [2000]).

To establish local convergence for the generic LCRS algorithm, we need several lemmas. In Lemma 3.1 we show that all local neighbors of $\mathbf{x}$ will be evaluated in the limit if $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ i.o.

LEMMA 3.1. *When the generic algorithm is applied to Problem* (1)*, if Conditions* 1 *and* 2 *hold and* $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ *i.o., then*

$$\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k) \text{ i.o.}\} = 0.$$

PROOF. By Condition 1, if $\mathbf{x} \in [\mathcal{LN}(\widehat{\mathbf{x}}_{k-1}^*) \setminus \mathcal{E}(k-1)]$, then $\Pr\{\mathbf{x} \in \mathcal{S}_k\} \geq \epsilon$ for some $\epsilon > 0$ for all $k = 1, 2, \ldots$. Let $M_k = \sum_{i=0}^{k} \mathcal{I}(\widehat{\mathbf{x}}_i^* = \mathbf{x})$ be the number of times that $\widehat{\mathbf{x}}_i^* = \mathbf{x}$ through iteration k, where $\mathcal{I}(\cdot)$ is the indicator function. Then for any $\mathbf{y} \in \mathcal{LN}(\mathbf{x})$ and any $k = 1, 2, \ldots,$

$$\Pr\{\mathbf{y} \notin \mathcal{E}(k)\} \leq (1-\epsilon)^{M_{k-1}}. \tag{2}$$

Since $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ i.o., then $M_k \to \infty$. Therefore

$$\lim_{k\to\infty} \Pr\{\mathbf{y} \notin \mathcal{E}(k)\} = 0.$$

Notice that

$$\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k)\} \leq \sum_{\text{all } \mathbf{y}\in\mathcal{LN}(\mathbf{x})} \Pr\{\mathbf{y} \notin \mathcal{E}(k)\}.$$

Since $|\mathcal{LN}(\mathbf{x})| \leq 2d + 1 < \infty$, where d is the dimension of $\mathbf{x}$, then by Eq. (2),

$$\lim_{k\to\infty} \Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k)\} = 0. \tag{3}$$

Furthermore, since $\mathcal{E}(i) = \cup_{j=1}^{i}\mathcal{E}_j$, we know that $\mathcal{E}(i) \subset \mathcal{E}(k)$ for all $k \geq i$. Thus $\mathcal{LN}(\mathbf{x}) \subset \mathcal{E}(i)$ implies $\mathcal{LN}(\mathbf{x}) \subset \mathcal{E}(k)$ for all $k \geq i$. Then $\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k)$ i.o. implies that $\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(i)$ for all $i = 1, 2, \ldots$. Therefore

$$\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k) \text{ i.o.}\} \leq \Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(i)\}$$

for every $i = 1, 2, \ldots$. Then by Eq. (3),

$$\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k) \text{ i.o.}\} \leq \lim_{i\to\infty} \Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(i)\} = 0.$$

This concludes the proof of the lemma. □

The next lemma shows that the sample-best solution $\widehat{\mathbf{x}}_k^*$ will become the best solution in $\mathcal{E}_k$ as $k \to \infty$.

LEMMA 3.2. *Let $\widehat{\mathbf{x}}_k^*$, $k = 0, 1, 2, \ldots$, be the sequence of solutions generated by the generic algorithm when applied to Problem* (1). *Suppose that Assumptions* 1 *and* 2 *are satisfied. If Conditions* 1 *and* 2 *hold, then*

$$\lim_{k\to\infty}\left[g(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right] = 0 \quad \text{w.p.1.}$$

PROOF. For any $\delta > 0$,

$$\begin{aligned}
&\Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta \text{ i.o.}\right\} \\
&\leq \Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \bar{G}_k(\widehat{\mathbf{x}}_k^*)\right| > \frac{\delta}{2} \text{ i.o.}\right\} + \Pr\left\{\left|\bar{G}_k(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \frac{\delta}{2} \text{ i.o.}\right\} \\
&= \Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \bar{G}_k(\widehat{\mathbf{x}}_k^*)\right| > \frac{\delta}{2} \text{ i.o.}\right\} + \Pr\left\{\left|\min_{\mathbf{y}\in\mathcal{E}_k} \bar{G}_k(\mathbf{y}) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \frac{\delta}{2} \text{ i.o.}\right\} \\
&\leq \Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \bar{G}_k(\widehat{\mathbf{x}}_k^*)\right| > \frac{\delta}{2} \text{ i.o.}\right\} + \Pr\left\{\max_{\mathbf{y}\in\mathcal{E}_k} \left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ i.o.}\right\} \qquad (4) \\
&\leq 2\Pr\left\{\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{E}_k \text{ i.o.}\right\}, \qquad (5)
\end{aligned}$$

where Eq. (4) holds according to Lemma 1 of Hong and Nelson [2006]. Notice that $\mathcal{E}(k) \subset \mathcal{E}(\infty)$ for all $k = 1, 2, \ldots$. Therefore

$$\text{Eq. (5)} \leq 2\Pr\left\{\mathcal{I}(\mathbf{y} \in \mathcal{E}_k)\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{E}(\infty) \text{ i.o.}\right\}. \tag{6}$$

Since $|\mathcal{E}(\infty)| < \infty$ w.p.1 by Condition 2, $|\mathcal{E}(k)|$ is nondecreasing in k and the increment is at least 1 if it increases, there exists an integer random variable K such that $\mathcal{E}(K) = \mathcal{E}(\infty)$ and $K < \infty$ w.p.1. Since $K < \infty$ w.p.1, we have that for any $\epsilon > 0$, there exists a constant positive integer K_ϵ such that $\Pr\{K \leq K_\epsilon\} \geq 1 - \epsilon$. Therefore

$$\Pr\{\mathcal{E}(K_\epsilon) = \mathcal{E}(\infty)\} \geq 1 - \epsilon.$$

Then,

$$\begin{aligned}
&\text{Eq. (6)}\\
&\leq 2\Pr\Big\{\mathcal{I}(\mathbf{y} \in \mathcal{E}_k)\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right|\\
&> \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{E}(\infty) \text{ i.o. and } \mathcal{E}(K_\epsilon) = \mathcal{E}(\infty)\Big\} + 2\epsilon\\
&\leq 2\Pr\left\{\mathcal{I}(\mathbf{y} \in \mathcal{E}_k)\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{E}(K_\epsilon) \text{ i.o.}\right\} + 2\epsilon. \qquad (7)
\end{aligned}$$

Let $\bar{G}_{K_\epsilon+1:k}(\mathbf{y})$ denote the sample mean of $G(\mathbf{y}, \psi)$ calculated with the samples obtained from iteration $K_\epsilon + 1$ to k for any $k > K_\epsilon$, namely,

$$\bar{G}_{K_\epsilon+1:k}(\mathbf{y}) = \begin{cases} \sum_{i=N_{K_\epsilon}(\mathbf{y})+1}^{N_k(\mathbf{y})} G_i(\mathbf{y})/\left[N_k(\mathbf{y}) - N_{K_\epsilon}(\mathbf{y})\right], & \text{if } N_k(\mathbf{y}) > N_{K_\epsilon}(\mathbf{y})\\ 0, & \text{if } N_k(\mathbf{y}) = N_{K_\epsilon}(\mathbf{y}) \end{cases}.$$

Then

$$\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| \leq \frac{N_{K_\epsilon}(\mathbf{y})}{N_k(\mathbf{y})}\left|\bar{G}_{K_\epsilon}(\mathbf{y}) - g(\mathbf{y})\right| + \frac{N_k(\mathbf{y}) - N_{K_\epsilon}(\mathbf{y})}{N_k(\mathbf{y})}\left|\bar{G}_{K_\epsilon+1:k}(\mathbf{y}) - g(\mathbf{y})\right|.$$

Since $\mathcal{E}(K_\epsilon)$ depends only on the observations obtained through iteration K_ϵ, and $N_k(\mathbf{y}) \to \infty$ if $\mathbf{y} \in \mathcal{E}_k$ i.o. by Condition 2, then, as $k \to \infty$,

$$\frac{N_{K_\epsilon}(\mathbf{y})}{N_k(\mathbf{y})}\left|\bar{G}_{K_\epsilon}(\mathbf{y}) - g(\mathbf{y})\right| \to 0.$$

Therefore

$$\begin{aligned}
&\text{Eq. (7)}\\
&\leq 2\Pr\Big\{\mathcal{I}(\mathbf{y} \in \mathcal{E}_k)\frac{N_k(\mathbf{y}) - N_{K_\epsilon}(\mathbf{y})}{N_k(\mathbf{y})}\left|\bar{G}_{K_\epsilon+1:k}(\mathbf{y}) - g(\mathbf{y})\right|\\
&\quad > \frac{\delta}{4} \text{ for some } \mathbf{y} \in \mathcal{E}(K_\epsilon) \text{ i.o.}\Big\} + 2\epsilon.
\end{aligned} \qquad (8)$$

Notice that when $k > K_\epsilon$, $0 \leq [N_k(\mathbf{y}) - N_{K_\epsilon}(\mathbf{y})]/N_k(\mathbf{y}) \leq 1$. Thus,

$$\begin{aligned}
&\text{Eq. (8)}\\
&\leq 2\Pr\left\{\mathcal{I}(\mathbf{y} \in \mathcal{E}_k)\left|\bar{G}_{K_\epsilon+1:k}(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{4} \text{ for some } \mathbf{y} \in \mathcal{E}(K_\epsilon) \text{ i.o.}\right\} + 2\epsilon. \qquad (9)
\end{aligned}$$

By Condition 1, $\bigcup_{i=1}^{k} \mathcal{C}_k \subset \mathcal{B}_k$. Then $\mathcal{E}(K_\epsilon) \subset \mathcal{B}_{K_\epsilon}$ and $\mathcal{B}_{K_\epsilon}$ is a finite and deterministic set. Since $\bar{G}_{K_\epsilon+1:k}(\mathbf{y})$ is independent of $\mathcal{E}(K_\epsilon)$, by Boole's inequality and

Assumption 2,

$$\begin{aligned}\text{Eq. (9)} &\le 2|\mathcal{B}_{K_\epsilon}| \sup_{\mathbf{y}\in\mathcal{B}_{K_\epsilon}} \Pr\left\{\mathcal{I}(\mathbf{y}\in\mathcal{E}_k)\left|\bar{G}_{K_\epsilon+1:k}(\mathbf{y})-g(\mathbf{y})\right| > \frac{\delta}{8}\text{ i.o.}\right\} + 2\epsilon \\ &= 2\epsilon.\end{aligned}$$

Since ϵ is arbitrary,

$$\Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta\text{ i.o.}\right\} = 0$$

for any $\delta > 0$, which implies the conclusion of the lemma. □

The next lemma states that $g(\widehat{\mathbf{x}}_k^*)$ converges w.p.1 as $k \to \infty$. In other words, for almost all realizations of the algorithm, $g(\widehat{\mathbf{x}}_k^*)$ converges to a limit as $k \to \infty$.

LEMMA 3.3. *Let $\widehat{\mathbf{x}}_k^*$, $k = 0, 1, 2, \ldots$, be a sequence of solutions generated by the generic algorithm when applied to Problem* (1). *Suppose that Assumptions* 1 *and* 2 *are satisfied. If Conditions* 1 *and* 2 *hold, then $g(\widehat{\mathbf{x}}_k^*)$, $k = 0, 1, 2, \ldots$, converges w.p.1.*

PROOF. By Condition 2, $\mathbf{x}_0 \in \mathcal{E}_k$ for all $k \ge 1$. Then by Lemma 3.2,

$$\Pr\left\{g(\widehat{\mathbf{x}}_k^*) > g(\mathbf{x}_0) + \delta\text{ i.o.}\right\} = 0$$

for all $\delta > 0$. Therefore by taking $\delta = \delta_0$, Assumption 1 implies that

$$\Pr\{\widehat{\mathbf{x}}_k^* \notin \mathcal{L}\text{ i.o.}\} = 0. \tag{10}$$

If $g(\mathbf{x}) = g(\mathbf{x}_0)$ for all $\mathbf{x} \in \mathcal{L}$, then Eq. (10) implies that $g(\widehat{\mathbf{x}}_k^*)$ converges to $g(\mathbf{x}_0)$ w.p.1. The conclusion of the lemma holds.

If $g(\mathbf{x}) \ne g(\mathbf{x}_0)$ for some $\mathbf{x} \in \mathcal{L}$, then let $\xi = \inf\{g(\mathbf{x}) - g(\mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{L}$ and $g(\mathbf{x}) - g(\mathbf{y}) > 0\}$. Since $\mathcal{L}$ is a finite set by Assumption 1, then $\xi > 0$. Let $0 < \delta < \xi$. By Lemma 3.2, we have

$$\Pr\left\{g(\widehat{\mathbf{x}}_k^*) > g(\widehat{\mathbf{x}}_{k-1}^*) + \delta\text{ i.o.}\right\} = 0. \tag{11}$$

Notice that for any $\mathbf{x}, \mathbf{y} \in \mathcal{L}$, $g(\mathbf{x}) > g(\mathbf{y})$ implies $g(\mathbf{x}) > g(\mathbf{y}) + \delta$ when $0 < \delta < \xi$. Then, by Eqs. (10) and (11),

$$\Pr\{g(\widehat{\mathbf{x}}_k^*) > g(\widehat{\mathbf{x}}_{k-1}^*)\text{ i.o.}\} = 0. \tag{12}$$

Also notice that for each realization of the algorithm where $g(\widehat{\mathbf{x}}_k^*)$ is not greater than $g(\widehat{\mathbf{x}}_{k+1}^*)$ i.o., we have $g(\widehat{\mathbf{x}}_k^*) \le g(\widehat{\mathbf{x}}_{k+1}^*)$ for all $k \ge K$, for some finite K. Since $g(\widehat{\mathbf{x}}_k^*)$ is bounded below ($\mathcal{L}$ is finite), $g(\widehat{\mathbf{x}}_k^*)$ converges for such a realization. By Eq. (12) this type of realization happens w.p.1. This concludes the proof of the lemma. □

Now we are ready to state and prove the local convergence of the generic algorithm.

THEOREM 3.4. *Let $\widehat{\mathbf{x}}_k^*$, $k = 0, 1, 2, \ldots$, be a sequence of solutions generated by the generic algorithm when applied to Problem* (1). *Suppose that Assumptions* 1 *and* 2 *are satisfied. If Conditions* 1 *and* 2 *hold, then* $\Pr\{\widehat{\mathbf{x}}_k^* \notin \mathcal{M}\text{ i.o.}\} = 0$.

Proof. In the proof of Lemma 3.3 we have shown that $\Pr\{\widehat{\mathbf{x}}_k^* \notin \mathcal{L} \text{ i.o.}\} = 0$. Since $\mathcal{L}$ is a finite set by Assumption 1, every subsequence of $\widehat{\mathbf{x}}_k^*$ has a further subsequence converging to a solution in $\mathcal{L}$ w.p.1. Therefore, proving Theorem 3.4 is equivalent to proving that $\mathbf{x} \in \mathcal{M}$ w.p.1 whenever $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ i.o.

On all sample paths for which $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ i.o., there exists an infinite subsequence $\widehat{\mathbf{x}}_{k_i}^*$, $i = 1, 2, \ldots$, such that $\widehat{\mathbf{x}}_{k_i}^* = \mathbf{x}$ for all i. Then Lemmas 3.2 and 3.3 ensure that

$$\lim_{i\to\infty}\left[g(\widehat{\mathbf{x}}_{k_i+1}^*) - \min_{\mathbf{y}\in\mathcal{E}_{k_i+1}} g(\mathbf{y})\right] = 0 \text{ w.p.1}$$

and $\lim_{i\to\infty} g(\widehat{\mathbf{x}}_{k_i+1}^*) = g(\mathbf{x})$. Therefore

$$\lim_{i\to\infty}\left\{\min_{\mathbf{y}\in\mathcal{E}_{k_i+1}} g(\mathbf{y})\right\} = g(\mathbf{x}) \text{ w.p.1.} \tag{13}$$

If $\widehat{\mathbf{x}}_k^* = \mathbf{x}$ i.o., then $\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}(k) \text{ i.o.}\} = 0$ by Lemma 3.1. Then Condition 2 guarantees that $\Pr\{\mathcal{LN}(\mathbf{x}) \not\subset \mathcal{E}_{k_i+1} \text{ i.o.}\} = 0$. Notice that $\min_{\mathbf{y}\in\mathcal{LN}(\mathbf{x})} g(\mathbf{y}) \geq \min_{\mathbf{y}\in\mathcal{E}_{k_i+1}} g(\mathbf{y})$ when $\mathcal{LN}(\mathbf{x}) \subset \mathcal{E}_{k_i+1}$. Therefore

$$\min_{\mathbf{y}\in\mathcal{LN}(\mathbf{x})} g(\mathbf{y}) \geq \lim_{i\to\infty}\left\{\min_{\mathbf{y}\in\mathcal{E}_{k_i+1}} g(\mathbf{y})\right\} \text{ w.p.1.} \tag{14}$$

Combining Eqs. (13) and (14) we have $g(\mathbf{x}) \leq \min_{\mathbf{y}\in\mathcal{LN}(\mathbf{x})} g(\mathbf{y})$ w.p.1. Therefore, $\mathbf{x} \in \mathcal{M}$ w.p.1. □

4. ASYMPTOTIC NORMALITY OF THE GENERIC ALGORITHM

In this section we analyze the limiting distribution of the estimated optimal objective function values generated by the generic algorithm. We show that they satisfy a central limit theorem under certain conditions.

To analyze the limiting distribution of the generic algorithm, we make the following additional assumptions.

Assumption 3. For any $\mathbf{x} \in \Theta$, the simulation observations $G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots$ are independent and identically distributed with mean $g(\mathbf{x})$ and variance $\sigma^2(\mathbf{x}) < \infty$.

Notice that Assumption 3 implies Assumption 2. This assumption enables us to apply the standard central limit theorem for $G_i(\mathbf{x})$, $i = 1, 2, \ldots$, for all $\mathbf{x} \in \Theta$.

Assumption 4. Problem (1) has only one local minimizer denoted as $\mathbf{x}^*$.

If there is more than a single local minimizer, the generic algorithm may converge to any one of them, and it is difficult to characterize the probability of converging to one in particular. When there is only one local minimizer, however, the algorithm will converge to it w.p.1. Although Assumption 4 is restrictive, the results we obtain in this section provide insights for problems with more than one local minimizer.

We will use the following notation: The symbol $\Rightarrow$ denotes "converges in distribution," the symbol $\sim$ denotes "has distribution," and $N(\mu, \sigma^2)$ denotes a normal random variable with mean μ and variance σ^2.

THEOREM 4.1. *Let $\widehat{\mathbf{x}}_k^*$, $k = 0, 1, 2, \dots$, be a sequence of solutions generated by the generic algorithm when applied to Problem* (1). *Suppose that Assumptions* 1, 3, *and* 4 *are satisfied. If Conditions* 1 *and* 2 *hold, and if there exists a deterministic sequence c_k, $k = 1, 2, \dots$, such that $N_k(\widehat{\mathbf{x}}_k^*)/c_k \to 1$ in probability as $k \to \infty$, then*

$$\sqrt{N_k(\widehat{\mathbf{x}}_k^*)} \cdot \left[\bar{G}_k(\widehat{\mathbf{x}}_k^*) - g(\mathbf{x}^*)\right] \Rightarrow \sigma(\mathbf{x}^*) \cdot N(0,1) \quad \text{as } k \to \infty.$$

PROOF. Let $Y_k = \sqrt{N_k(\widehat{\mathbf{x}}_k^*)} \cdot \left[\bar{G}_k(\widehat{\mathbf{x}}_k^*) - g(\mathbf{x}^*)\right]$. Then

$$\begin{aligned} Y_k = {} & \mathcal{I}(\widehat{\mathbf{x}}_k^* = \mathbf{x}^*)\sqrt{N_k(\mathbf{x}^*)} \cdot \left[\bar{G}_k(\mathbf{x}^*) - g(\mathbf{x}^*)\right] \\ & + \left(1 - \mathcal{I}(\widehat{\mathbf{x}}_k^* = \mathbf{x}^*)\right)\sqrt{N_k(\widehat{\mathbf{x}}_k^*)} \cdot \left[\bar{G}_k(\widehat{\mathbf{x}}_k^*) - g(\mathbf{x}^*)\right]. \end{aligned} \tag{15}$$

Since $\mathcal{M} = \{\mathbf{x}^*\}$ by Assumption 4, Theorem 3.4 implies that $\Pr\{\widehat{\mathbf{x}}_k^* \neq \mathbf{x}^* \text{ i.o.}\} = 0$. Therefore, $\mathcal{I}(\widehat{\mathbf{x}}_k^* = \mathbf{x}^*) \to 1$ w.p.1. By the continuous mapping theorem and the converging together lemma [Durrett 1995], it suffices to show that

$$\sqrt{N_k(\mathbf{x}^*)} \cdot \left[\bar{G}_k(\mathbf{x}^*) - g(\mathbf{x}^*)\right] \Rightarrow \sigma(\mathbf{x}^*) \cdot N(0,1) \quad \text{as } k \to \infty. \tag{16}$$

Since $\Pr\{\widehat{\mathbf{x}}_k^* \neq \mathbf{x}^* \text{ i.o.}\} = 0$, we have $N_k(\widehat{\mathbf{x}}_k^*)/N_k(\mathbf{x}^*) \to 1$ w.p.1, and $N_k(\mathbf{x}^*)/c_k \to 1$ in probability. Then Eq. (16) follows directly from the random index central limit theorem [Durrett 1995]. □

Remarks.

(1) Theorem 4.1 and its proof are similar to Theorem 4.1 of Andradóttir [1999], which analyzes the limiting distribution of globally convergent random-search algorithms. However, her result only applies to problems with finite Θ, while Theorem 4.1 applies to problems with finite and countably infinite Θ.

(2) Theorem 4.1 requires that $N_k(\widehat{\mathbf{x}}_k^*)/c_k \to 1$ in probability, which is often satisfied. Many random-search algorithms use a prespecified deterministic sequence to determine the sample size of the current sample-best solution. For instance, Hong and Nelson [2006] suggest using an equal simulation-allocation rule, namely, $N_k(\mathbf{x}) = N_k$ for all $\mathbf{x} \in \mathcal{E}_k$, to take advantage of common random numbers. Then $N_k(\widehat{\mathbf{x}}_k^*)/N_k = 1$ for all $k \geq 1$.

Theorem 4.1 provides a means to compare different DOvS algorithms. Since the rate of convergence of the algorithm depends only on $N_k(\widehat{\mathbf{x}}_k^*)$, the percentage of the total number of observations allocated to $\widehat{\mathbf{x}}_k^*$ provides a natural measure of the efficiency of the algorithms. Let T_k be the total number of simulation observations used by the algorithm through iteration k. Notice that T_k is often a random variable, since it is determined by how many solutions are sampled through iteration k. The ratio $\gamma_k = N_k(\widehat{\mathbf{x}}_k^*)/T_k$ measures the fraction of effort spent on the sample-best solution through iteration k. Let $\gamma_\infty = \lim_{k\to\infty} N_k(\widehat{\mathbf{x}}_k^*)/T_k$ be the asymptotic ratio. Notice that higher γ_∞ means that the algorithm is asymptotically more efficient in estimating the optimal objective value. If γ_∞ exists, then we may use it compare the asymptotic efficiency of algorithms. One such instance is presented in Section 6.

5. DESIGNING ESTIMATION SCHEMES WHEN Θ IS COUNTABLY INFINITE

Condition 2 requires that $|\mathcal{E}(\infty)| < \infty$ w.p.1., which is satisfied if Θ is a finite set, since $\mathcal{E}(\infty)$ is a subset of Θ. However, when Θ is countably infinite, this requirement needs some additional care. In this section, we introduce a general approach that guarantees $|\mathcal{E}(\infty)| < \infty$ w.p.1.

We make the following additional assumption on $G_i(\mathbf{x})$.

Assumption 5. For any positive constant δ and any $\mathbf{x} \in \Theta$ there exist positive numbers r^* and δ^* such that for all $r \geq r^*$ and for all $0 < \delta \leq \delta^*$

$$\mathrm{P}\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| > \delta\right] \leq \lambda(r, \delta),$$

where $\lambda(r, \delta)$ is a strictly decreasing function of r and $\lambda(r, \delta) \to 0$ as $r \to \infty$.

Assumption 5 has also been used in Hong and Nelson [2006] to design the COMPASS algorithm for solving DOvS problems with countably infinite feasible regions. They showed that $\lambda(r, \delta) = c/r\delta^2$ for some constant $c > 0$ if $G_i(\mathbf{x})$ has a uniformly bounded asymptotic variance for all $\mathbf{x} \in \Theta$, and $\lambda(r, \delta) = e^{-I_\delta r}$ for some constant $I_\delta > 0$ if $G_i(\mathbf{x})$ has a uniformly bounded large-deviation rate function for all $\mathbf{x} \in \Theta$.

We substitute the following for Condition 2 on the estimation scheme.

Condition 3. The estimation scheme satisfies the following requirements:

(1) $\mathcal{E}_k$ is a subset of $\mathcal{S}(k)$,
(2) $\mathcal{E}_k$ contains at least $\mathbf{x}_0$, $\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1}) \cap \mathcal{E}(k-1)$ and $\mathcal{S}_k$,
(3) $a_k(\mathbf{x})$ is allocated such that $\min_{\mathbf{x}\in\mathcal{E}_k} N_k(\mathbf{x}) \geq r_k$ for all $k = 1, 2, \ldots$, where r_k is a deterministic sequence satisfying $r_k \geq 1$ for all $k = 1, 2, \ldots$ and $\sum_{k=1}^{\infty} |\mathcal{B}_k|\lambda(r_k, \delta) < \infty$ for any $0 < \delta \leq \delta^*$, and
(4) if $\bigcup_{k=0}^{\infty}\{\widehat{\mathbf{x}}^*_k\}$ is a finite set, then $\bigcup_{k=1}^{\infty} \mathcal{C}_k$ is also a finite set.

The two approaches to constructing $\mathcal{C}_k$ and $\mathcal{B}_k$, introduced in Section 2, both satisfy the third and fourth requirements of Condition 3. In particular, in both approaches, $|\mathcal{B}_k| \leq (c + 2k\Delta)^d$ for some positive constant c. Then the third requirement of Condition 3 can be satisfied if $k^{d+1}\lambda(r_k, \delta) \to 0$ as $n \to \infty$.

Compared to Condition 2, Condition 3 does not directly require $|\mathcal{E}(\infty)| < \infty$ w.p.1. However, it dictates more about how the simulation observations are allocated in each iteration. In the next lemma, we show that Condition 3 implies Condition 2 if Assumption 5 is satisfied.

LEMMA 5.1. *Suppose that the generic algorithm is applied to solve Problem* (1) *and Assumptions* 1 *and* 5 *are satisfied. If Conditions* 1 *and* 3 *hold, then Condition 2 holds as well.*

Proof. Comparing Conditions 2 and 3, we only need to prove $|\mathcal{E}(\infty)| < \infty$ w.p.1. By the proof of Eq. (5), for any $\delta > 0$,

$$\Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta\right\} \leq 2\Pr\left\{\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{E}_k\right\}. \tag{17}$$

By Condition 1, $\bigcup_{i=1}^k \mathcal{C}_k \subset \mathcal{B}_k$. Therefore $\bigcup_{i=0}^k S_k \subset \mathcal{B}_k$, which implies that $\mathcal{E}_k \subset \mathcal{B}_k$. Now suppose that on iteration k, all solutions in $\mathcal{B}_k$ have at least r_k observations, not just those in the estimation set. Then for any $\delta < \min(\delta_0, \delta^*)$, where δ_0 and δ^* are defined in Assumptions 1 and 5 respectively, by Eq. (17),

$$\begin{aligned}
&\Pr\left\{\left|g(\widehat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta\right\}\\
&\leq\ 2\Pr\left\{\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2} \text{ for some } \mathbf{y} \in \mathcal{B}_k\right\}\\
&\leq\ 2|\mathcal{B}_k| \sup_{\mathbf{y}\in\mathcal{B}_k} \Pr\left\{\left|\bar{G}_k(\mathbf{y}) - g(\mathbf{y})\right| > \frac{\delta}{2}\right\}\\
&\leq\ 2|\mathcal{B}_k|\lambda(r_k, \delta/2),
\end{aligned}$$

where the last equation follows from Assumption 5. Then, by Condition 3

$$\sum_{k=0}^{\infty} \Pr\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta\right\} \leq \sum_{k=0}^{\infty} |\mathcal{B}_k| \lambda\left(r_k, \delta/2\right) < \infty.$$

Thus, the first Borel-Cantelli lemma Billingsley [1995] implies that

$$\Pr\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})\right| > \delta \text{ i.o.}\right\} = 0. \tag{18}$$

Since $\mathbf{x}_0 \in \mathcal{E}_k$ for all $k = 1, 2, \ldots$ and $g(\mathbf{x}_0) \geq \min_{\mathbf{y}\in\mathcal{E}_k} g(\mathbf{y})$, Eq. (18) implies that

$$\Pr\left\{g(\hat{\mathbf{x}}_k^*) > g(\mathbf{x}_0) + \delta \text{ i.o.}\right\} = 0.$$

Therefore $\Pr\{\hat{\mathbf{x}}_k^* \notin \mathcal{L} \text{ i.o.}\} = 0$. Notice that $\mathcal{L}$ is a finite set by Assumption 1. Then $\bigcup_{k=0}^{\infty}\{\widehat{\mathbf{x}}_k^*\}$ is a finite set w.p.1, which implies that $\bigcup_{k=1}^{\infty} \mathcal{C}_k$ is a finite set w.p.1 by the fourth requirement of Condition 3. Since $\mathcal{E}(k) \subset \mathcal{S}(k)$ and $\mathcal{S}(k) \subset \bigcup_{i=1}^k \mathcal{C}_i$, then $\mathcal{E}(\infty) \subset \bigcup_{k=1}^{\infty} \mathcal{C}_k$. Therefore $|\mathcal{E}(\infty)| < \infty$ w.p.1. □

6. REVISED COMPASS ALGORITHM

In this section we apply the conditions developed in the article to revise the COMPASS algorithm of Hong and Nelson [2006]. We show that the efficiency of the algorithm can be further improved.

On each iteration, COMPASS creates a most promising area $\mathcal{C}_k$, which is the sampling neighborhood on iteration k based on the location of the current sample-best solution and all other previously sampled solutions, and then samples a new set of solutions from $\mathcal{C}_k$ uniformly. All the solutions sampled through this iteration are estimated, that is, $\mathcal{E}_k = \mathcal{S}(k)$, according to a simulation-allocation rule. One can easily check that Conditions 1 and 2 are both satisfied by the COMPASS algorithm. Therefore, it is locally convergent.

As discussed in Hong and Nelson [2006], one of the advantages of COMPASS is the flexibility of $\mathcal{C}_k$: It is large at the beginning of the search, which enables COMPASS to explore a large area, and becomes smaller and smaller as more and more solutions are sampled, enabling COMPASS to concentrate on local comparisons. Notice that this advantage is only related to the sampling scheme of COMPASS. Its estimation scheme, which is to evaluate *all* sampled solutions, may be improved. According to Condition 2, only a subset of the sampled solutions are required to be estimated on each iteration to guarantee convergence; estimating all sampled solutions is certainly not necessary. In this section we revise the COMPASS algorithm, showing that the revised version satisfies Conditions 1 and 2, and thus has the desired asymptotic properties. We then compare the asymptotic efficiencies and numerical performances of the original COMPASS and the revised one.

6.1 The Algorithm

Let Θ_k be the set of feasible solutions in consideration on iteration k of COMPASS. Then $\Theta_k = \Theta$ if Θ is finite, and Θ_k=$\Theta \cap \mathcal{H}_k$ if Θ is countably infinite, where $\mathcal{H}_k$ is a bounding hyperbox on iteration k (see Hong and Nelson [2006] for the definition of $\mathcal{H}_k$). The most promising area $\mathcal{C}_k$ is defined as

$$\mathcal{C}_k = \left\{\mathbf{x} \in \Theta_k : \|\mathbf{x} - \widehat{\mathbf{x}}^*_{k-1}\| \le \|\mathbf{x} - \mathbf{y}\|,\ \forall\, \mathbf{y} \in \mathcal{S}(k-1)\right\}.$$

It includes all solutions that are closer to the current sample-best solution $\widehat{\mathbf{x}}^*_{k-1}$ than to other previously sampled solutions through iteration $k-1$. It is equivalent to the definition

$$\mathcal{C}_k = \left\{\mathbf{x} \in \Theta_k : (\widehat{\mathbf{x}}^*_{k-1} - \mathbf{y})' \left(\mathbf{x} - \frac{\widehat{\mathbf{x}}^*_{k-1} + \mathbf{y}}{2}\right) \ge 0,\ \forall\, \mathbf{y} \in \mathcal{S}(k-1)\right\}. \tag{19}$$

Notice that some of the previously sampled solutions in $\mathcal{S}(k-1)$ may be redundant in the sense that $\mathcal{C}_k$ remains the same if those solutions are removed from $\mathcal{S}(k-1)$. For example, if all solutions in $\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1})$ have been sampled, then the set $\mathcal{C}_k$, determined by $\mathcal{S}(k-1)$, and $\widehat{\mathbf{x}}^*_{k-1}$ is the same as the set

$$\left\{\mathbf{x} \in \Theta_k : (\widehat{\mathbf{x}}^*_{k-1} - \mathbf{y})' \left(\mathbf{x} - \frac{\widehat{\mathbf{x}}^*_{k-1} + \mathbf{y}}{2}\right) \ge 0,\ \forall\, \mathbf{y} \in \mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1})\right\}.$$

In this case all sampled solutions that are not in $\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1})$ are redundant for determining $\mathcal{C}_k$. Let $\mathcal{P}_k$ be the set of previously sampled solutions that *actually* form $\mathcal{C}_k$, including $\widehat{\mathbf{x}}^*_{k-1}$. Then

$$\mathcal{C}_k = \left\{\mathbf{x} \in \Theta_k : (\widehat{\mathbf{x}}^*_{k-1} - \mathbf{y})' \left(\mathbf{x} - \frac{\widehat{\mathbf{x}}^*_{k-1} + \mathbf{y}}{2}\right) \ge 0,\ \forall\, \mathbf{y} \in \mathcal{P}_k\right\}.$$

Notice that $\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1}) \cap \mathcal{S}(k-1)$ is a subset of $\mathcal{P}_k$. Then, by Condition 2, instead of estimating all sampled solutions on each iteration of COMPASS, we need only estimate newly sampled solutions, $\mathbf{x}_0$, and solutions in $\mathcal{P}_k$, namely, $\mathcal{E}_k = \mathcal{S}(k) \cup \{\mathbf{x}_0\} \cup \mathcal{P}_k$.

In practice, however, $\mathcal{P}_k$ is difficult to find when the dimension of the problem is larger than two. To solve this problem we design a simple algorithm to find a

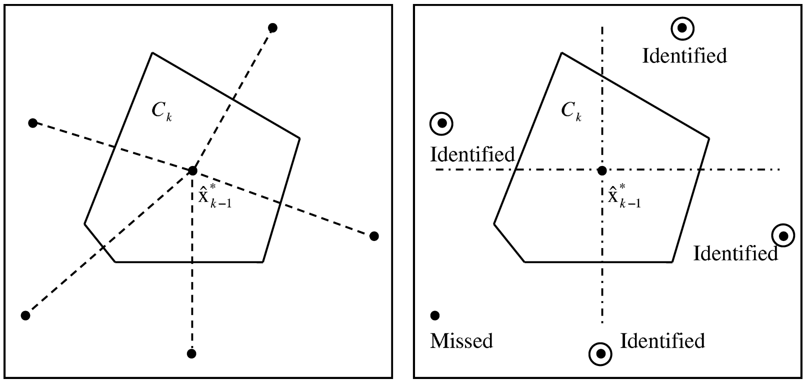

Fig. 1. Constructing $\mathcal{P}'_k$ from $\mathcal{C}_k$.

set $\mathcal{P}'_k$ that approximates $\mathcal{P}_k$ and still guarantees that $[\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1}) \cap \mathcal{S}(k-1)] \subset \mathcal{P}'_k$.

As noted in Hong and Nelson [2006], if Θ is a set of d-dimensional integer solutions in a convex set Φ, then $\mathcal{C}_k$ is also a set of d-dimensional integer solutions in a convex and compact set $\tilde{\mathcal{C}}_k$, where

$$\tilde{\mathcal{C}}_k = \left\{ \mathbf{x} \in \Phi_k : (\widehat{\mathbf{x}}^*_{k-1} - \mathbf{y})' \left(\mathbf{x} - \frac{\widehat{\mathbf{x}}^*_{k-1} + \mathbf{y}}{2} \right) \geq 0, \ \forall\, \mathbf{y} \in \mathcal{S}(k-1) \right\}$$

and $\Phi_k = \Phi$ if Θ is finite and $\Phi_k = \Phi \cap \mathcal{H}_k$ if Θ is countably infinite. Then we use the following algorithm to find $\mathcal{P}'_k$.

Algorithm Construction of $\mathcal{P}'_k$

Step 1. Construct $\tilde{\mathcal{C}}_k$ using $\widehat{\mathbf{x}}^*_{k-1}$ and $\mathcal{S}(k-1)$.

Step 2. For every $i = 1, 2, \ldots, d$ draw a line passing through $\widehat{\mathbf{x}}^*_{k-1}$ and parallel to the x_i axis. The line intersects with the boundary of $\tilde{\mathcal{C}}_k$ at two points $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ since $\tilde{\mathcal{C}}_k$ is convex and compact. Then

$$\mathcal{P}'_k = \bigcup_{i=1}^{d} \left\{ \mathbf{y} \in \mathcal{S}(k-1) : (\widehat{\mathbf{x}}^*_{k-1} - \mathbf{y})' \left(\mathbf{x}_{ij} - \frac{\widehat{\mathbf{x}}^*_{k-1} + \mathbf{y}}{2} \right) = 0 \text{ for either } j = 1 \text{ or } 2 \right\}.$$

Step 3. Let $\mathcal{P}'_k = \mathcal{P}'_k \cup \{\widehat{\mathbf{x}}^*_{k-1}\}$.

Notice that $\mathcal{P}'_k$ is a subset of $\mathcal{P}_k$ and $\mathcal{LN}(\widehat{\mathbf{x}}^*_{k-1}) \cap \mathcal{S}(k-1) \subset \mathcal{P}'_k$. Therefore, if we let $\mathcal{E}_k = \mathcal{S}_k \cup \{\mathbf{x}_0\} \cup \mathcal{P}'_k$, Condition 2 is still satisfied. Figure 1 is an example of applying the construction algorithm to a two-dimensional problem. In the example, the most promising area $\mathcal{C}_k$ is formed by the current sample-best solution, and five sampled solutions which form $\mathcal{P}_k$ in the left panel of Figure 1. Redundant solutions and constraints are not plotted. The right panel shows how to construct $\mathcal{P}'_k$. Notice that only four points from $\mathcal{P}_k$ are identified in this example.

This leads to the following revised COMPASS algorithm.

Algorithm Revised COMPASS

Step 0. Let $\mathbf{x}_0 \in \Theta$ be the starting solution provided by the user. Set the iteration count $k = 0$. Let $N_0(\mathbf{x}) = 0$ and $\bar{G}_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \Theta$, let the current sample-best solution be $\widehat{\mathbf{x}}^*_0 = \mathbf{x}_0$ and let the set of sampled solutions be $\mathcal{S}_0 = \{\mathbf{x}_0\}$.

Step 1. Let $k = k + 1$. Construct the most promising area $\mathcal{C}_k$ of $\widehat{\mathbf{x}}^*_{k-1}$ according to Eq. (19). Sample a set of m_k solutions, $\mathcal{S}_k$, uniformly from $\mathcal{C}_k$.

Step 2. Construct $\mathcal{P}'_k$ and let $\mathcal{E}_k = \mathcal{S}_k \cup \{\mathbf{x}_0\} \cup \mathcal{P}'_k$. For every $\mathbf{x} \in \mathcal{E}_k$, allocate $a_k(\mathbf{x})$ observations based on the simulation-allocation rule of COMPASS and $\bar{G}_k(\mathbf{x})$. For all $\mathbf{x} \notin \mathcal{E}_k$, let $N_k(\mathbf{x}) = N_{k-1}(\mathbf{x})$ and $\bar{G}_k(\mathbf{x}) = \bar{G}_{k-1}(\mathbf{x})$.

Step 3. Let $\widehat{\mathbf{x}}^*_k = \operatorname{argmin}_{\mathbf{x}\in\mathcal{E}_k} \bar{G}_k(\mathbf{x})$. If there is more than one solution having the smallest sample mean $\min_{\mathbf{x}\in\mathcal{E}_k} \bar{G}_k(\mathbf{x})$, then any such solution can be chosen as $\widehat{\mathbf{x}}^*_k$. Go to Step 1.

If Θ is a finite set and Assumptions 1 and 2 are satisfied, then the local convergence of the revised COMPASS algorithm is a direct result of Theorem 3.4. If Θ is countably infinite and Assumptions 1, 2, and 5 are satisfied, then the local convergence of the algorithm follows from Lemma 5.1 and Theorem 3.4.

As pointed out in Section 4, the asymptotic efficiencies of random-search algorithms can be compared by $\gamma_\infty = \lim_{k\to\infty} N_k(\widehat{\mathbf{x}}^*_k)/T_k$, which is the asymptotic percentage of the simulation effort allocated to the sample-best solution. Suppose that there exists only one local minimizer. Then both algorithms converge to the same optimal solution w.p.1. We further assume that both algorithms use the equal simulation-allocation rule, which allocates simulation observations such that all solutions in $\mathcal{E}_k$ have N_k observations to take advantage of common random numbers.

Let $\mathcal{E}_\infty = \lim_{k\to\infty} \mathcal{E}_k$ if the limit exits. Then $|\mathcal{E}_\infty| < \infty$ w.p.1, since $|\mathcal{E}(\infty)| < \infty$ w.p.1 by Condition 2, and $\mathcal{E}_\infty$ contains all solutions that are estimated infinitely often. Under the equal simulation-allocation rule, all solutions in $\mathcal{E}_\infty$ have the same number of observations and all solutions that are not in $\mathcal{E}_\infty$ have only a finite number of observations in the limit. Therefore, $\gamma_\infty = 1/|\mathcal{E}_\infty|$.

For the revised COMPASS algorithm, it is clear that $\mathcal{E}_\infty = \{\mathbf{x}_0\} \cup \mathcal{LN}(\mathbf{x}^*)$ w.p.1. Therefore $|\mathcal{E}_\infty| = 1 + |\mathcal{LN}(\mathbf{x}^*)|$ w.p.1. For the COMPASS algorithm, $\mathcal{E}_\infty = \mathcal{S}(\infty)$, where $\mathcal{S}(\infty)$ is a random set. However, $\mathcal{S}(\infty)$ contains at least $\mathbf{x}_0$ and all solutions in $\mathcal{LN}(\mathbf{x}^*)$, namely, $|\mathcal{S}(\infty)| \geq 1 + |\mathcal{LN}(\mathbf{x}^*)|$. Therefore, the revised COMPASS algorithm is asymptotically more efficient than COMPASS.

6.2 Numerical Example

In this subsection, we use an example to compare the numerical performances of COMPASS and revised COMPASS. The numerical example shows that revised COMPASS is indeed more efficient than COMPASS in finding a local minimizer.

The following is a practical problem from one of the largest air cargo terminals in the world. One of the terminal's critical resources is shipping/receiving docks. Export/import cargos are delivered/picked up by trucks of freight forwarders at the docks. There are four main types of cargo that are delivered/picked up at the docks: pallet bulk cargo, general bulk cargo, perishable

Table I. Arrival and Service Rates

Cargo Type	Arrival Rate (/hour)	Average Service Time (minutes)
pallet bulk	$\lambda_1 = 52.8$	$1/\mu_1 = 67$
general bulk	$\lambda_2 = 11.7$	$1/\mu_2 = 46$
perishable	$\lambda_3 = 13.0$	$1/\mu_3 = 92$
prepacked	$\lambda_4 = 22.5$	$1/\mu_1 = 34$

cargo, and prepacked cargo. Different types of cargo use various material handling systems and require different operations. Therefore, the terminal needs to decide how to allocate a limited number of docks to these four types of cargo to minimize the average waiting time of trucks. This problem is especially important during the peak season (the months prior to Christmas and New Year), since the terminal is operated near designed capacity. The current allocation scheme was based on workload balancing. Basically, docks were allocated such that the average workload of each is about the same. We are interested in verifying if the current scheme is optimal and finding the optimal allocation scheme if not.

In practice, we use historical data to construct the arrival processes and service-time distributions of trucks, and they are time dependent. But to illustrate the revised COMPASS algorithm for this problem, we use the following simple model: We assume that there are totally M docks, the arrival processes of trucks for pallet bulk cargo, general bulk cargo, perishable cargo, and prepacked cargo are stationary Poisson processes with rates λ_1, λ_2, λ_3, and λ_4, the service times of all four types of cargo are exponentially distributed with rates μ_1, μ_2, μ_3, and μ_4, and we minimize the long-run average waiting time of trucks. We let x_1, x_2, x_3, and x_4 denote the number of docks assigned to the four types of cargos. Notice that we can write $x_4 = M - x_1 - x_2 - x_3$. Therefore, it is a three-dimensional optimization problem. To ensure stability of queues, we require that $\lambda_i < x_i \mu_i$ for all $i = 1, 2$, and, 3. We can build a model to simulate the system, given x_1, x_2, and x_3, and find the long-run average waiting time. Therefore, this issue can be solved as a DOvS problem.

There are two reasons why we solve the simplified problem instead of the more realistic one: First, the complexities for solving this problem and solving that with realistic data are similar. If revised COMPASS can solve this problem well, then we believe it can also solve the more realistic problem well. Second, the long-run average waiting time of trucks, given x_1, x_2, and x_3, can be calculated explicitly in this simpler model. Therefore, we know the actual performance of the revised COMPASS algorithm. In the following experiments, we set $M = 111$. The arrival and service rates are set according to Table I.

For each simulation replication we set the warm-up period as 100 hours, then simulate the system for 50 hours after warm-up and calculate the average waiting time of trucks. We use the equal simulation-allocation rule and let $N_k = \min\{5, \lceil 5(\log k)^{1.01} \rceil\}$, which is the same rule used in Hong and Nelson [2006]. We set the number of solutions sampled on each iteration as 5. We run COMPASS and revised COMPASS 100 times for this problem and record all the information. The results are summarized next.

Table II. Local Minimizers

Solutions	x_1	x_2	x_3	Average Waiting Time (minutes)
1	63	11	22	10.24
2	63	10	23	10.93
3	64	10	22	11.09

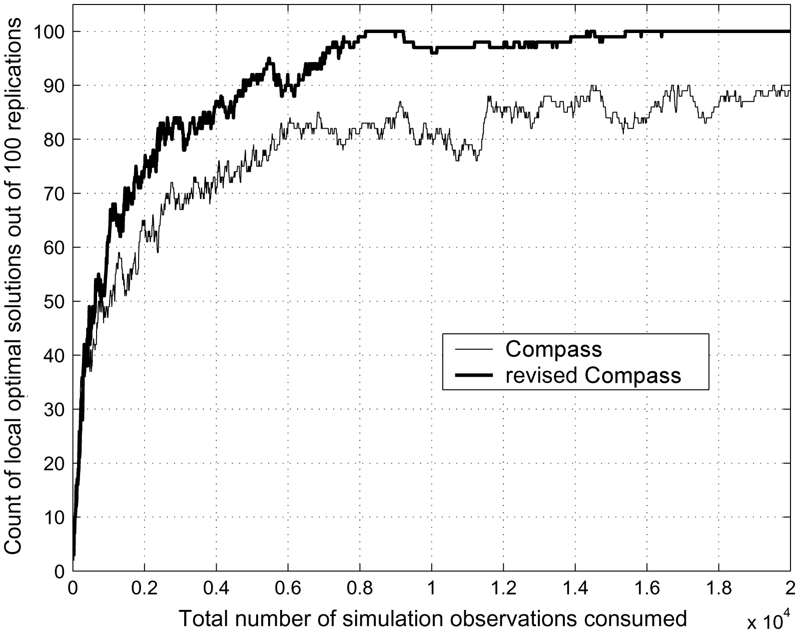

Fig. 2. Performance of COMPASS and revised COMPASS.

The algorithms find 3 local minimizers to the problem. They are listed in Table II. If we use the terminal's approach to balancing the workload, we have $x_1 = 65$, $x_2 = 10$, and $x_3 = 22$ with an average waiting time of 12.64 minutes. We see that all 3 local minimizers are significantly better than this solution. The optimal solutions show that it is advantageous to let the average workload per dock of pallet bulk cargo be slightly higher than the overall average. The result is also consistent with our experiments with real data.

To compare the performance of COMPASS and revised COMPASS, we look at two plots. In Figure 2 we compare how good the two algorithms are at finding local minimizers. Any point on the two curves represents the number out of 100 trials that the current sample-best solution $\widehat{\mathbf{x}}_k^*$ is a local minimizer at a given total number of simulation observations consumed. From Figure 2 we see clearly that revised COMPASS has a higher probability of finding local minimizers than COMPASS when given the same amount of simulation effort. Therefore revised COMPASS has better performance.

The major difference between COMPASS and revised COMPASS is that the former allocates simulation observations to all sampled solutions on each iteration, while the latter only allocates to a selected subset of sampled solutions. Therefore, compared to COMPASS, revised COMPASS allocates more

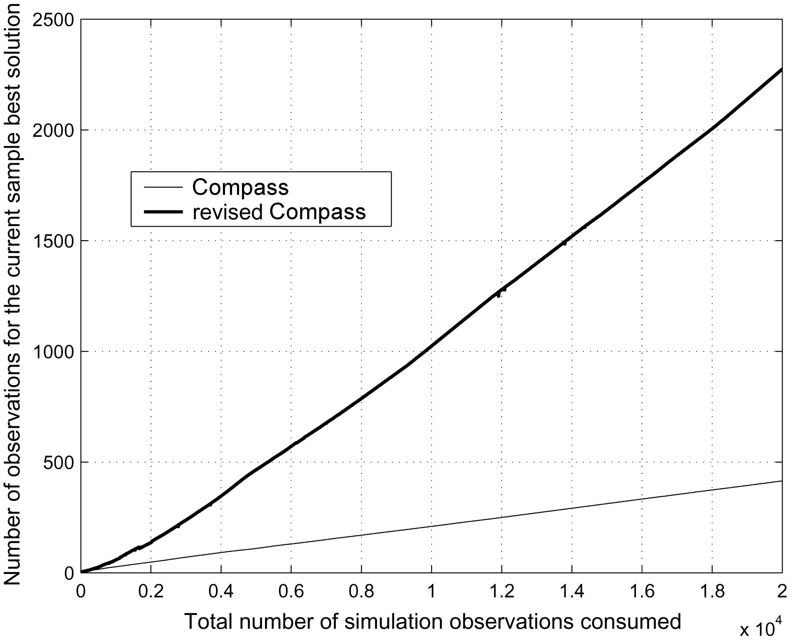

Fig. 3. Number of observations taken of the current sample-best solution.

simulation observations to the current sample-best solution and its local neighbors to ensure a better comparison in the local neighborhood. To see this effect we plot in Figure 3 the average number of simulation observations allocated to the current sample-best solution for both algorithms when the total number of simulation observations is given. We can see that as the total number of simulation observations increases, the difference between the two curves becomes more significant. Therefore, the noise level in function evaluations in revised COMPASS becomes significantly lower than in COMPASS. This in turn increases the probability that the current sample-best solution is a local optimal solution.

ACKNOWLEDGMENTS

The authors acknowledge the help of the editors and referees in revising the article.

REFERENCES

Alrefaei, M. H. and Andradóttir, S. 1999. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Manage. Sci. 45*, 748–764.

Andradóttir, S. 2006. Simulation optimization with countably infinite feasible regions: Efficiency and convergence. *ACM Trans. Model. Comput. Simul. 16*, 357–374.

Andradóttir, S. 1999. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Trans. Model. Comput. Simul. 9*, 349–380.

Andradóttir, S. 2002. Simulation optimization: Integrating research and practice. *INFORMS Journal on Computing*, 14, 216–219.

April, J., Better, M., Glover, F., and Kelly, J. 2004. New advances and applications for marrying simulation and optimization. In *Proceedings of the Winter Simulation Conference*, R .G. Ingalls et al., Eds. IEEE, Piscataway, NJ. 80–86.

Baras, J. S. 2003. Modeling and simulation of telecommunication networks for control and management. In *Proceedings of the Winter Simulation Conference*, S. Chick et al. Eds. IEEE, Piscataway, NJ. 431–440.

Bechhofer, R. E., Santner, T. J., and Goldsman, D. M. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.

Billingsley, P. 1995. *Probability and Measure*, 3rd ed. Wiley, New York.

Boesel, J., Nelson, B. L., and Kim, S.-H. 2003. Using ranking and selection to 'clean up' after simulation optimization. *Oper. Res. 51*, 814-825.

Chen, C.-H., Lin, J., Yücesan, E., and Chick, S. E. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynam. Syst. Theory Appl. 10*, 251–270.

Chick, S. E. and Inoue, K. 2001. New two-stage and sequential procedures for selecting the best simulated system. *Oper. Res. 49*, 732–743.

Durrett, R. 1995. *Probability: Theory and Examples*, 2nd ed. Duxbury Press, Belmont, CA.

Fu, M. C. 2002. Optimization for simulation: Theory vs. practice. *INFORMS J. Comput. 14*, 192–215.

Gong, W.-B., Ho, Y.-C., and Zhai, W. 1999. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM J. Optim. 10*, 384–404.

Hong, L. J. and Nelson, B. L. 2006. Discrete optimization via simulation using COMPASS. *Oper. Res. 54*, 115–129.

Hong, L. J. and Nelson, B. L. 2007. Selecting the best system when systems are revealed sequentially. *IIE Trans. 39*, 723–734.

Kleywegt, A., Shapiro, A., and Homem-de-Mello, T. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM J. Opt. 12*, 479–502.

Law, A. M. and Kelton, W. D. 2000. *Simulation Modeling and Analysis*, 3rd ed. McGraw-Hill, New York.

Nelson, B. L., Swann, J., Goldsman, D., and Song, W. 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Oper. Res. 49*, 950–963.

Shi, L. and Ólafsson, S. 2000. Nested partitions method for stochastic optimization. *Methodol. Comput. Appl. Probab. 2*, 271–291.

Truong, T. H. and Azadivar, F. 2003. Simulation based optimization for supply chain configuration design. In *Proceedings of the Winter Simulation Conference*, S. Chick et al., Eds. IEEE, Piscataway, NJ. 1268–1275.

Vogt, H. 2004. A new method to determine the tool count of a semiconductor factory using FabSim. In *Proceedings of the Winter Simulation Conference*, R .G. Ingalls et al., Eds. IEEE, Piscataway, NJ. 1925–1929.

Wieland, F. and Holden, T. C. 2003. Targeting aviation delay through simulation optimization. In *Proceedings of the Winter Simulation Conference*, S. Chick et al., Eds. IEEE, Piscataway, NJ. 578–584.

Wolsey, L. A. 1998. *Integer Programming*. John Wiley, New York.

Yan, D. and Mukai, H. 1992. Stochastic discrete optimization. *SIAM J. Control Optim. 30*, 594–612.

Received February 2006; revised March 2007; accepted March 2007