OPERATIONS RESEARCH
Vol. 54, No. 1, January–February 2006, pp. 115–129
ISSN 0030-364X | EISSN 1526-5463 | 06 | 5401 | 0115

DOI 10.1287/opre.1050.0237

Discrete Optimization via Simulation Using COMPASS

L. Jeff Hong
Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China, hongl@ust.hk

Barry L. Nelson
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208-3119, nelsonb@northwestern.edu

We propose an optimization-via-simulation algorithm, called COMPASS, for use when the performance measure is estimated via a stochastic, discrete-event simulation, and the decision variables are integer ordered. We prove that COMPASS converges to the set of local optimal solutions with probability 1 for both terminating and steady-state simulation, and for both fully constrained problems and partially constrained or unconstrained problems under mild conditions.

Subject classifications: simulation, design of experiments: optimization via simulation; programming, stochastic: adaptive random search.
Area of review: Simulation.
History: Received January 2004; revisions received June 2004, October 2004; accepted October 2004.

1. Introduction

Every day, simulation users build simulation models to analyze manufacturing, financial, communication, computer, and service systems. One important feature of simulation experiments is that users can choose different system settings to try to improve the performance of their systems. Therefore, it is natural to search for settings that optimize the system performance. This is called optimization via simulation (OvS). OvS is different from deterministic optimization and the typical stochastic programming problem because there is no explicit form of the objective function, and function evaluations are stochastic and computationally expensive.

Recently, research interests have shifted from OvS problems with continuous decision variables to OvS problems with discrete decision variables (called discrete OvS or DOvS). DOvS algorithms include the stochastic ruler method, with specific implementations being SR1992 (Yan and Mukai 1992) and SR2001 (Alrefaei and Andradóttir 2001), Andradóttir's random search methods AR1995 and AR1996 (Andradóttir 1995, 1996), simulated annealing algorithm SA (Alrefaei and Andradóttir 1999), stochastic comparison method SC (Gong et al. 1999), nested partitions, with specific implementations being NP2000 (Shi and Ólafsson 2000) and NP2003 (Pichitlamken and Nelson 2003), sample average approximation method (Kleywegt et al. 2001, Homem-de-Mello 2000), and simultaneous perturbation stochastic approximation method (Gerencsér et al. 1999). For a summary of OvS research and practice, see Fu (2002), Andradóttir (1998), and Ólafsson and Kim (2002).

Many DOvS algorithms are based on random search. In this paper, we provide an algorithm called *convergent optimization via most-promising-area stochastic search* (COMPASS), which is also based on random search. However, COMPASS has a unique neighborhood structure that, to the best of our knowledge, has never been proposed in the literature. The neighborhood is defined as the most promising area at each iteration, and the most promising area is fully adaptive (this differs from nested partitions in which the "most promising region" is chosen from a predetermined collection of nested sets). Initially, we consider all feasible solutions to be equally promising, so the most promising area is the set of all feasible solutions. Once some solutions have been visited (simulated), the promising index of every feasible solution is the sample mean performance of the visited solution that is closest to it. The most promising area is defined as the set of feasible solutions that are at least as close to the current sample best solution as they are to other visited solutions. COMPASS generates candidate solutions uniformly from the most promising area in each iteration and then assigns simulation experiments to every solution in the set of visited solutions according to a simulation-allocation rule. We show that COMPASS converges with probability 1 to a set of local optimal solutions and that it has robust finite-time performance.

This paper is organized as follows. We first provide a literature review of random search algorithms that have been applied to DOvS in §2. In §§3 and 4, we present the COMPASS algorithms for fully constrained DOvS problems and partially constrained or unconstrained DOvS problems. Section 5 shows how to construct and sample

from the most promising area. Numerical examples are provided in §6, followed by conclusions in §7.

2. Background

Many DOvS algorithms are based on random search. There are at least three ways to categorize random search algorithms: by neighborhood structure, by the number of feasible solutions, and by convergence properties.

Random search algorithms typically generate candidate solutions from the neighborhood of a selected solution in each iteration. Some algorithms have a fixed neighborhood structure, while others change their neighborhood structure based on the information gained through the optimization process. Among the random search algorithms mentioned above, SR1992, SR2001, AR1995, AR1996, SA, and SC all have fixed neighborhood structures. Whenever the algorithm revisits a solution, the candidate solutions are generated from the same neighborhood. NP2000 and NP2003, however, have an adaptive neighborhood structure. Roughly speaking, they use a large neighborhood at the beginning of the search when knowledge about the location of good solutions is limited, and the neighborhood shrinks as more and more information on the objective function is revealed. When good solutions are clustered together, which is often true in simulation, algorithms with an adaptive neighborhood structure often work better than algorithms with a fixed neighborhood structure.

With respect to the number of feasible solutions, random search algorithms can be divided into two categories: algorithms that solve problems with a finite number of feasible solutions, and algorithms that solve problems with a countably infinite number of feasible solutions. In this paper, the feasible solutions are the integer points in a feasible region that is defined by a collection of constraints. When the decision variables of a DOvS problem are fully constrained, then the problem has a finite number of feasible solutions; when the decision variables are partially constrained or unconstrained, then the problem has a countably infinite number of feasible solutions. One might argue that all local optimal solutions of any reasonable OvS problem should be achievable. Therefore, adding an upper bound and a lower bound for every decision variable can turn a partially constrained or unconstrained problem into a fully constrained problem. However, finding reasonable bounds such that the optimal solutions are within the feasible region, without making the search space overwhelmingly large, is a very difficult problem. Therefore, it is worthwhile to design algorithms that efficiently solve partially constrained or unconstrained DOvS problems. Among the random search algorithms in the literature, AR1995 and SR2001 are, to the best of our knowledge, the only algorithms that solve problems with a countably infinite number of feasible solutions. AR1995 solves only one-dimensional problems, and SR2001 requires strong conditions on the distributions of simulation outputs at each feasible solution.

The convergence properties of DOvS algorithms can be divided into three categories: no guaranteed convergence, locally convergent, and globally convergent. Most algorithms used in commercial software are heuristics, such as tabu search and scatter search in OptQuest[1] (Glover et al. 1996), that provide no convergence guarantee. These algorithms typically evaluate the objective function by averaging over a small (often fixed) number of replications, and then treat the average as deterministic. When the variances of the objective function values are high and they differ significantly over the feasible region, the heuristics can be misled. Methods have been proposed to enhance the performance of heuristics by providing a statistical inference on all visited solutions at the end of search; these include ordinal optimization (Ho et al. 2000) and ranking and selection (Boesel et al. 2003).

Random search algorithms in the literature are often globally convergent, including SR1992, SR2001, AR1996, SA, SC, NP2000, and NP2003. Any globally convergent algorithm that assumes no structure on the objective function requires visiting every feasible solution infinitely often to guarantee convergence. However, when the number of feasible solutions is large or the computational budget is low, requiring the algorithm to visit a large number of solutions, if not every solution, is unreasonable. Therefore, in this paper we focus on designing a locally convergent algorithm for both fully constrained and partially constrained or unconstrained problems. Moreover, if global convergence is required, we can always make a locally convergent algorithm globally convergent by adding a pure random search component or by restarting.

Convergent random search algorithms often prove their convergence by constructing a Markov chain over the solution space. SR1992 and SC build time-inhomogeneous Markov chains on the solution space and then show that the Markov chain degenerates to the set of optimal solutions. To satisfy Markov properties, SR1992 and SC both discard simulation observations from previous iterations and require more and more simulation observations per solution as the iteration count increases. Because simulation experiments are computationally expensive, we believe it makes sense to accumulate observations instead of discarding them.

SR2001, AR1995, AR1996, SA, and NP2000 also build Markov chains. However, their Markov chains are time homogeneous, and they are shown to visit optimal solutions most often. Past information is used in these algorithms through counts of how many visits a solution receives. Therefore, the optimization process does not slow down when the iteration count gets large. However, the function evaluations are still discarded to satisfy the Markov property.

Andradóttir (1999) provides a scheme to accelerate the convergence of DOvS algorithms by accumulating previous observations. Her idea is to keep the original search algorithm but report the solution with the best aggregated sample mean, the sample mean calculated from all simulation

observations taken from the solution, as the best. She shows that the empirical performance of DOvS algorithms can be improved significantly by doing so. When her scheme is applied to Markov-chain-based algorithms, however, the aggregated sample means are used only to report the best solution; they are not used to direct the random search so as to retain the Markov property. NP2003, on the other hand, exploits Andradóttir's scheme while also using aggregated sample means to determine the current best solution and to direct the search. Therefore, it fully utilizes the past information.

To use past information, either through visit counts or by aggregated sample means, one needs to keep a list of all visited solutions, and when a new solution is generated one needs to check if it has already been visited. The storage and checking cost can be high if the algorithm visits a large number of solutions. However, the computational cost and storage cost are often small compared to the cost of conducting simulation experiments in practical real-world, discrete-event simulation models. Therefore, we believe that the past information should be used in any practical DOvS algorithm.

The COMPASS algorithms provided in this paper have an adaptive neighborhood structure, solve both fully constrained and partially constrained or unconstrained problems, and converge to the set of local optimal solutions.

3. COMPASS for Fully Constrained DOvS

Consider the following DOvS problem:

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

where $\Theta = \Phi \cap \mathcal{Z}^d$, Φ is a closed and bounded set in $\Re^d$, and $\mathcal{Z}^d$ is the set of d-dimensional vectors with integer elements. To avoid triviality, we assume that Θ is nonempty. Therefore, $0 < |\Theta| < \infty$, where $|\cdot|$ denotes the cardinality of a set. Note that Problem (1) is fully constrained because Φ is bounded. The quantity ψ represents the stochastic input to the simulation, and its distribution might 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 $g(\mathbf{x}) = \mathrm{E}_{\psi}[G(\mathbf{x},\psi)]$ and assume that $g(\mathbf{x})$ cannot be evaluated easily (or at all) but 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 $G_i(\mathbf{x})$. We make the following assumption about $G_i(\mathbf{x})$.

Assumption 1. *For every* $\mathbf{x}\in\Theta$, *we have*

$$\mathrm{P}\left[\lim_{r\to\infty}\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) = g(\mathbf{x})\right] = 1.$$

Assumption 1 implies that the sample mean of $G(\mathbf{x},\psi)$ is an appropriate estimator of $g(\mathbf{x})$. Note that if $G_i(\mathbf{x})$, $i = 1, 2, \ldots$, are independent and identically distributed, Assumption 1 becomes the *strong law of large numbers*. If $G_i(\mathbf{x})$, $i = 1, 2, \ldots$, are ergodic, Assumption 1 becomes the *ergodic theorem*. Most simulation output satisfies Assumption 1.

Let $\mathcal{N}(\mathbf{x}) = \{\mathbf{y}\colon\ \mathbf{y}\in\Theta \text{ and } \|\mathbf{x}-\mathbf{y}\| = 1\}$ be the local neighborhood of $\mathbf{x}\in\Theta$, where $\|\mathbf{x}-\mathbf{y}\|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. We define $\mathbf{x}$ as a *local minimizer* if $\mathbf{x}\in\Theta$ and either $\mathcal{N}(\mathbf{x}) = \varnothing$ or $g(\mathbf{x}) \leqslant g(\mathbf{y})$ for all $\mathbf{y}\in\mathcal{N}(\mathbf{x})$. Let $\mathcal{M}$ denote the set of local minimizers of the function g in Θ.

In the COMPASS algorithm, we use $\mathcal{V}_k$ to denote the set of all solutions visited through iteration k, and use $\hat{\mathbf{x}}_k^*$ to denote the solution with the smallest aggregated sample mean among all $\mathbf{x}\in\mathcal{V}_k$. If there is more than one solution having the smallest aggregated sample mean, then we select $\hat{\mathbf{x}}_k^*$ randomly from the set of solutions having the smallest aggregated sample mean. At the end of iteration k, we construct $\mathcal{C}_k = \{\mathbf{x}\colon \mathbf{x}\in\Theta \text{ and } \|\mathbf{x}-\hat{\mathbf{x}}_k^*\| \leqslant \|\mathbf{x}-\mathbf{y}\| \ \forall\,\mathbf{y}\in\mathcal{V}_k \text{ and } \mathbf{y}\neq\hat{\mathbf{x}}_k^*\}$, the most promising area at iteration k. The set $\mathcal{C}_k$ includes all feasible solutions that are at least as close to $\hat{\mathbf{x}}_k^*$ as to other solutions in $\mathcal{V}_k$. At iteration $k+1$, we will sample m solutions uniformly from $\mathcal{C}_k$, where m can be set to any positive integer. Note that $\mathcal{C}_k$ is never empty because $\hat{\mathbf{x}}_k^*$ is always in $\mathcal{C}_k$, and we do not require the m solutions to be unique.

The COMPASS algorithm needs a simulation-allocation rule (SAR) to allocate simulation observations to solutions in $\mathcal{V}_k$ at each iteration. Let $a_k(\mathbf{x})$ be the additional observations allocated to $\mathbf{x}$ at iteration k as determined by the SAR; $a_k(\mathbf{x})$ might depend on all *past* information such as $\mathcal{V}_k$ and $G_i(\mathbf{x})$, $i = 1, 2, \ldots$, for all $\mathbf{x}\in\mathcal{V}_k$. Then, $N_k(\mathbf{x}) = \sum_{i=0}^{k} a_i(\mathbf{x})$ denotes the total number of observations on solution $\mathbf{x}$ at iteration k for every $\mathbf{x}\in\mathcal{V}_k$. We use $\bar{G}_k(\mathbf{x})$ to denote the sample mean of all $N_k(\mathbf{x})$ observations of $G(\mathbf{x},\psi)$ at iteration k.

Algorithm 1 (COMPASS for Fully Constrained DOvS)

Step 0. Set iteration count $k = 0$. Find $\mathbf{x}_0\in\Theta$, set $\mathcal{V}_0 = \{\mathbf{x}_0\}$ and $\hat{\mathbf{x}}_0^* = \mathbf{x}_0$. Determine $a_0(\mathbf{x}_0)$ according to the SAR. Take $a_0(\mathbf{x}_0)$ observations from $\mathbf{x}_0$, set $N_0(\mathbf{x}_0) = a_0(\mathbf{x}_0)$, and calculate $\bar{G}_0(\mathbf{x}_0)$. Let $\mathcal{C}_0 = \Theta$.

Step 1. Let $k = k+1$. Sample $\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{km}$ uniformly and independently from $\mathcal{C}_{k-1}$. Let $\mathcal{V}_k = \mathcal{V}_{k-1} \cup \{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{km}\}$. Determine $a_k(\mathbf{x})$ according to the SAR for every $\mathbf{x}$ in $\mathcal{V}_k$. For all $\mathbf{x}\in\mathcal{V}_k$, take $a_k(\mathbf{x})$ observations, and update $N_k(\mathbf{x})$ and $\bar{G}_k(\mathbf{x})$.

Step 2. Let $\hat{\mathbf{x}}_k^* = \arg\min_{\mathbf{x}\in\mathcal{V}_k} \bar{G}_k(\mathbf{x})$. Construct $\mathcal{C}_k$ and go to Step 1.

Remarks. 1. In Step 1 of Algorithm 1, we sample m solutions independently from $\mathcal{C}_{k-1}$, which means that repeats in $\{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{km}\}$ are allowed. Indeed, the chance of getting repeats typically increases as the iteration count increases, because the most promising areas are getting smaller as the iteration count increases. If $\mathcal{C}_{k-1}$ becomes a singleton, then all m solutions sampled are $\hat{\mathbf{x}}_{k-1}^*$.

2. We do not specify a stopping rule for Algorithm 1 to prove its convergence. In practice, the algorithm can be stopped whenever all of the computational budget is consumed, or when $\hat{\mathbf{x}}_k^*$ does not change for many iterations and all solutions in its local neighborhood have been visited.

3. Algorithm 1 needs to store $N_k(\mathbf{x})$ and $\bar{G}_k(\mathbf{x})$ (or equivalently $N_k(\mathbf{x})$ and $\sum_{i=1}^{N_k(\mathbf{x})} G_i(\mathbf{x})$) for all visited solutions and construct the most promising area in each iteration. Compared to the computational effort used to conduct simulation experiments, however, the storage and construction costs are often low. Note that no previously visited solutions can be sampled in Step 1 except the current best. Therefore, if a candidate solution is not the current best, then it is a new solution. This property allows us to save the computational cost of checking if a candidate solution has already been visited as required in AR1995, AR1996, SA, NP2000, and NP2003.

4. Algorithm 1 may not visit every solution in Θ, even if the computational budget is infinite, which is different from globally convergent DOvS algorithms.

To make Algorithm 1 converge, we make the following assumption on the SAR.

ASSUMPTION 2. *The SAR guarantees that* $a_k(\mathbf{x}) \geqslant 1$ *if* $\mathbf{x}$ *is a newly visited solution at iteration* k $(\mathbf{x} \in \mathscr{V}_k \backslash \mathscr{V}_{k-1})$, *and* $\lim_{k\to\infty} N_k(\mathbf{x}) = +\infty$ *for all visited solutions* $(\mathbf{x} \in \bigcup_{k=0}^{\infty} \mathscr{V}_k)$.

The simplest SAR satisfying Assumption 2 is an equal SAR that sets $N_k(\mathbf{x}) = N_k$ for all $\mathbf{x} \in \mathscr{V}_k$ and $N_k \to \infty$ as $k \to \infty$ (for instance, $N_k = k$). One can also design adaptive SARs that satisfy Assumption 2 to make Step 2 of Algorithm 1 more efficient; see Chen et al. (2000) and Hong and Nelson (2004) for related work.

We have the following convergence theorem.

THEOREM 1. *If Assumptions* 1 *and* 2 *are satisfied, then the infinite sequence* $\{\hat{\mathbf{x}}_0^*, \hat{\mathbf{x}}_1^*, \ldots\}$ *generated by Algorithm* 1 *converges with probability* 1 *to the set* $\mathscr{M}$ *in the sense that* $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o.\} = 0$.

REMARKS. 1. The conclusion $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o.\} = 0$ is an almost sure convergence. It implies that, with probability 1, $\hat{\mathbf{x}}_k^*$ is not a local optimal solution only finitely many times. It also implies that $\mathrm{P}\{\hat{\mathbf{x}}_k^* \in \mathscr{M}\} \to 1$ as $k \to \infty$.

2. Assumption 1 is the only distributional assumption we need for the simulation output to guarantee local convergence for Algorithm 1. Therefore, the conclusion of Theorem 1 holds for both terminating and steady-state simulation problems. Moreover, we do not need independence between $G_i(\mathbf{x})$ and $G_j(\mathbf{y})$, $\mathbf{x} \neq \mathbf{y}$. Therefore, common random numbers (CRN) can be used to make Step 2 of Algorithm 1 more efficient, especially when an equal SAR is used, because CRN typically increase the probability that the sample best solution is actually the best solution visited through that iteration.

3. If $\mathscr{M}$ is a singleton, then the element in $\mathscr{M}$ is the global optimal solution, and Theorem 1 guarantees the global convergence of Algorithm 1.

To prove Theorem 1 we need the following lemma. The lemma is proved in the appendix.

LEMMA 1. *For real numbers* $a_1, a_2, \ldots, a_n$ *and* $b_1, b_2, \ldots, b_n$,

$$\left|\min_{i=1,\ldots,n} a_i - \min_{i=1,\ldots,n} b_i\right| \leqslant \max_{i=1,\ldots,n} |a_i - b_i|.$$

Now we can prove Theorem 1.

PROOF OF THEOREM 1. For any infinite sequence $\{\mathscr{V}_0, \mathscr{V}_1, \ldots\}$ generated by Algorithm 1, $\mathscr{V}_\infty = \bigcup_{k=0}^{\infty} \mathscr{V}_k$ exists and $\mathscr{V}_\infty \subset \Theta$ because $\mathscr{V}_k \subset \mathscr{V}_{k+1}$ and $\mathscr{V}_k \subset \Theta$ for all $k = 0, 1, \ldots$; and because Θ is a finite set,

$$\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o.\} = \sum_{A\subset\Theta} \mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o. \mid \mathscr{V}_\infty = A\}\,\mathrm{P}\{\mathscr{V}_\infty = A\}.$$

Therefore, proving $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o.\} = 0$ is equivalent to proving $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o. \mid \mathscr{V}_\infty = A\} = 0$ for all nonempty $A \subset \Theta$ such that $\mathrm{P}\{\mathscr{V}_\infty = A\} > 0$.

We first prove that

$$\mathrm{P}\left\{\lim_{k\to\infty} g(\hat{\mathbf{x}}_k^*) = \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x}) \mid \mathscr{V}_\infty = A\right\} = 1. \tag{2}$$

Equation (2) is equivalent to the statement that for any $\epsilon > 0$,

$$\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x})\right| \geqslant \epsilon \ i.o. \mid \mathscr{V}_\infty = A\right\} = 0. \tag{3}$$

Note that

$$\begin{aligned}
&\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x})\right| \geqslant \epsilon \ i.o. \mid \mathscr{V}_\infty = A\right\} \\
&\quad\leqslant \mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*) - \bar{G}_k(\hat{\mathbf{x}}_k^*)\right| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathscr{V}_\infty = A\right\} \\
&\qquad+ \mathrm{P}\left\{\left|\bar{G}_k(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x})\right| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathscr{V}_\infty = A\right\}. \qquad (4)
\end{aligned}$$

Consider the sequence $\{\mathscr{V}_0, \mathscr{V}_1, \ldots\}$ generated by Algorithm 1. Because $|\mathscr{V}_\infty| < \infty$, there exists a $K > 0$ (possibly depending on the sequence) such that $\mathscr{V}_k = \mathscr{V}_\infty$ for all $k \geqslant K$. Therefore,

$$\begin{aligned}
(4) &\leqslant \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \text{ for some } \mathbf{x}\in\mathscr{V}_\infty \mid \mathscr{V}_\infty = A\right\} \\
&\quad+ \mathrm{P}\left\{\left|\min_{\mathbf{x}\in\mathscr{V}_\infty} \bar{G}_k(\mathbf{x}) - \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x})\right| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathscr{V}_\infty = A\right\} \\
&\leqslant \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \text{ for some } \mathbf{x}\in\mathscr{V}_\infty \mid \mathscr{V}_\infty = A\right\} \\
&\quad+ \mathrm{P}\left\{\max_{\mathbf{x}\in\mathscr{V}_\infty} |\bar{G}_k(\mathbf{x}) - g(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathscr{V}_\infty = A\right\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{by Lemma 1} \\
&\leqslant 2\mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \text{ for some } \mathbf{x}\in\mathscr{V}_\infty \mid \mathscr{V}_\infty = A\right\} \\
&\leqslant 2\sum_{\mathbf{x}\in A} \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathscr{V}_\infty = A\right\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{by the Bonferroni inequality} \\
&= 2\sum_{\mathbf{x}\in A} \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathbf{x} \in \mathscr{V}_\infty\right\}. \qquad (5)
\end{aligned}$$

The last equality is true because the simulation observations taken from $\mathbf{x}$ when $k \geqslant K$ depend only on $\mathbf{x} \in \mathscr{V}_\infty$, not on $\mathscr{V}_\infty = A$. By Assumption 2, $N_k(\mathbf{x}) \to \infty$ if $\mathbf{x} \in \mathscr{V}_\infty$, and by Assumption 1, $\lim_{k\to\infty} \bar{G}_k(\mathbf{x}) = g(\mathbf{x})$ with probability 1 if $N_k(\mathbf{x}) \to \infty$. Therefore,

$$\mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \ i.o. \,\middle|\, \mathbf{x} \in \mathscr{V}_\infty\right\} = 0.$$

Because $|A| < \infty$, this implies that (5) is 0, which proves (2).

We now prove $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o. \mid \mathscr{V}_\infty = A\} = 0$ for any nonempty set $A \subset \Theta$ such that $\mathrm{P}\{\mathscr{V}_\infty = A\} > 0$. If $\hat{\mathbf{x}}_k^* \notin \mathscr{M}$ $i.o.$, then there exists an $\mathbf{x} \in A$ and $\mathbf{x} \notin \mathscr{M}$ such that $\hat{\mathbf{x}}_k^* = \mathbf{x}$ $i.o.$ Because $\mathbf{x} \notin \mathscr{M}$, there exists a $\mathbf{y} \in \mathscr{N}(\mathbf{x})$ such that $g(\mathbf{y}) < g(\mathbf{x})$. Note that because $\|\mathbf{y} - \mathbf{x}\| = 1 \leqslant \|\mathbf{y} - \mathbf{z}\|$ for all $\mathbf{z} \neq \mathbf{y}$, we have

$$\mathrm{P}\{\mathbf{y} \in \mathscr{V}_{k+1} \mid \hat{\mathbf{x}}_k^* = \mathbf{x} \text{ and } \mathbf{y} \notin \mathscr{V}_k\} \geqslant \frac{1}{|\Theta|} > 0 \tag{6}$$

for any iteration $k+1$. Because $\hat{\mathbf{x}}_k^* = \mathbf{x}$ $i.o.$,

$$\mathrm{P}\{\mathbf{y} \in A \mid \mathscr{V}_\infty = A \text{ and } \hat{\mathbf{x}}_k^* = \mathbf{x} \ i.o.\} = 1.$$

Note that if $\mathbf{y} \in A$, then $g(\mathbf{x}) \neq \min_{\mathbf{z}\in A} g(\mathbf{z})$ because $g(\mathbf{y}) < g(\mathbf{x})$. Then, by Equation (2) we know that, with probability 1, $\hat{\mathbf{x}}_k^*$ can only equal $\mathbf{x}$ finitely many times. This is a contradiction. Therefore, $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o. \mid \mathscr{V}_\infty = A\} = 0$ for any nonempty set $A \subset \Theta$ such that $\mathrm{P}\{\mathscr{V}_\infty = A\} > 0$. This proves the theorem. □

Remark. In Algorithm 1 we assume uniform sampling from the most promising area in each iteration. However, it is clear from (6) that Theorem 1 holds if

$$\mathrm{P}\{\mathbf{y} \in \mathscr{V}_{k+1} \mid \hat{\mathbf{x}}_k^* = \mathbf{x} \text{ and } \mathbf{y} \notin \mathscr{V}_k\} \geqslant c \quad \forall \mathbf{y} \in \mathscr{N}(\mathbf{x}), \tag{7}$$

where c is some positive constant. Therefore, one may consider other sampling distributions that satisfy Equation (7) and have some desirable properties, such as biasing sampling toward solutions closer to the current best.

4. COMPASS for Partially Constrained or Unconstrained DOvS

In Problem (1), if Φ is a closed but unbounded set in $\Re^d$, then we call the problem a partially constrained or unconstrained problem. If $\Phi = \Re^d$, then the problem is unconstrained; otherwise, it is partially constrained. We make the following assumption on $G(\mathbf{x}, \psi)$.

Assumption 3. *For any positive constant ϵ and any $\mathbf{x} \in \Theta$, there exist positive numbers r^* and ϵ^* such that for all $r \geqslant r^*$ and for all $0 < \epsilon \leqslant \epsilon^*$,*

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

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

Consider the following two special cases of Assumption 3. Let $\sigma^2(\mathbf{x}) = \lim_{r\to\infty} r\mathrm{Var}[\sum_{i=1}^{r} G_i(\mathbf{x})/r]$ be the asymptotic variance of $G_i(\mathbf{x})$, $i = 1, 2, \ldots$. Assume that $\sigma^2(\mathbf{x})$ exists for all $\mathbf{x} \in \Theta$, that $\sup_{\mathbf{x}\in\Theta} \sigma^2(\mathbf{x}) < \infty$, and that $\mathrm{E}[G_i(\mathbf{x})] = g(\mathbf{x})$. Then, for any $\delta > 0$ there exists a positive number r^* such that

$$r\mathrm{Var}\left[\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x})\right] \leqslant \sup_{\mathbf{x}\in\Theta} \sigma^2(\mathbf{x}) + \delta$$

for all $r \geqslant r^*$ and for all $\mathbf{x} \in \Theta$. By Chebyshev's inequality, we have for $r \geqslant r^*$ and for every $\mathbf{x} \in \Theta$,

$$\mathrm{P}\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| > \epsilon\right] \leqslant \frac{\sup_{\mathbf{x}\in\Theta} \sigma^2(\mathbf{x}) + \delta}{r\epsilon^2} \equiv \lambda(r, \epsilon). \tag{8}$$

The conditions in this case are satisfied by any i.i.d. sequence with finite and uniformly bounded variance, and they are also satisfied by many stochastic processes.

Here is the second case: Let $G_1(\mathbf{x}), G_2(\mathbf{x}), \ldots$ satisfy the large deviation principle (LDP, see Dembo and Zeitouni 1998),

$$\limsup_{r\to\infty} \frac{1}{r} \log \mathrm{P}\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| \geqslant \epsilon\right] \leqslant -I(\mathbf{x}, \epsilon), \tag{9}$$

where $0 < I(\mathbf{x}, \epsilon) < +\infty$ for all $\epsilon > 0$ and $\mathbf{x} \in \Theta$. If $\inf_{\mathbf{x}\in\Theta} I(\mathbf{x}, \epsilon) > 0$, then there exists a positive number r^* such that for all $r \geqslant r^*$,

$$\begin{aligned}\frac{1}{r} \log \mathrm{P}&\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| > \epsilon\right] \\ &\leqslant \frac{1}{r} \log \mathrm{P}\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| \geqslant \epsilon\right] \leqslant -\frac{\inf_{\mathbf{x}\in\Theta} I(\mathbf{x}, \epsilon)}{2}\end{aligned}$$

for all $\mathbf{x} \in \Theta$. Therefore, for any $\epsilon > 0$ and $r \geqslant r^*$ and for any $\mathbf{x} \in \Theta$,

$$\begin{aligned}\mathrm{P}\left[\left|\frac{1}{r}\sum_{i=1}^{r} G_i(\mathbf{x}) - g(\mathbf{x})\right| > \epsilon\right] &\leqslant \exp\left[-r\frac{\inf_{\mathbf{x}\in\Theta} I(\mathbf{x}, \epsilon)}{2}\right] \\ &\equiv \lambda(r, \epsilon).\end{aligned} \tag{10}$$

Equation (9) is satisfied by many i.i.d. sequences, for instance, random variables whose moment-generating functions are finite in a neighborhood of 0. Many dependent sequences also satisfy Equation (9), including finite-state Markov chains.

To establish local convergence, we make the following assumption on the objective function.

Assumption 4. *For the user provided starting point $\mathbf{x}_0$, there exists a compact set Π and a positive constant δ such that $\mathbf{x}_0 \in \Pi \cap \Theta$ and $g(\mathbf{x}) \geqslant g(\mathbf{x}_0) + \delta$ for any $\mathbf{x} \in \Pi^c \cap \Theta$.*

Many DOvS problems have a benchmark system setting, say $\mathbf{x}_0$, which is often the current system setting. All solutions beyond some (unknown) distance from the benchmark setting will typically be inferior to the benchmark. Therefore, Assumption 4 is satisfied.

The basic idea of the COMPASS algorithm for partially constrained or unconstrained DOvS problems can be described as follows: We construct a hyper-rectangle $\mathscr{B}_0$, $\mathscr{B}_0 = \prod_{i=1}^{d}[\underline{b}_0^{(i)}, \bar{b}_0^{(i)}]$, around the initial solution $\mathbf{x}_0$. Then, $\Phi \cap \mathscr{B}_0$ is a convex and compact set, and we can apply Algorithm 1. We may expand the rectangle adaptively in each iteration so it can grow to include local optimal solutions. In each iteration of COMPASS, we sample a set of candidate solutions from the most promising area, as in Algorithm 1, and calculate the distance from each solution to each boundary plane of the rectangle. For instance, the distances from candidate solution $\mathbf{x} = [x^{(1)}, \ldots, x^{(d)}]$ to boundary planes $\underline{b}_{k-1}^{(i)}$ and $\bar{b}_{k-1}^{(i)}$ are $x^{(i)} - \underline{b}_{k-1}^{(i)}$ and $\bar{b}_{k-1}^{(i)} - x^{(i)}$, respectively. Whenever a candidate solution is within $\Delta^{(i)}$ distance from a boundary, we expand the boundary such that the distance to the boundary is $\Delta^{(i)}$. The $\Delta^{(i)}$, $i = 1, 2, \ldots, d$, are user-provided positive numbers that control the rate that $\mathscr{B}_k$ grows with the restriction that $\Delta^{(i)} \geqslant 1$ for all $i = 1, 2, \ldots, d$. The algorithm is as follows.

Algorithm 2 (COMPASS for Partially Constrained or Unconstrained DOvS)

Step 0. Set iteration counter $k = 0$. Find $\mathbf{x}_0 \in \Theta$, set $\mathscr{V}_0 = \{\mathbf{x}_0\}$ and $\hat{\mathbf{x}}_0^* = \mathbf{x}_0$. Determine $a_0(\mathbf{x}_0)$ according to the SAR. Take $a_0(\mathbf{x}_0)$ observations from $\mathbf{x}_0$, set $N_0(\mathbf{x}_0) = a_0(\mathbf{x}_0)$, and calculate $\bar{G}(\mathbf{x}_0)$. Construct $\mathscr{B}_0 = \prod_{i=1}^{d}[\underline{b}_0^{(i)}, \bar{b}_0^{(i)}]$ such that $\underline{b}_0^{(i)} < x_0^{(i)} < \bar{b}_0^{(i)}$ for $i = 1, 2, \ldots, d$. Let $\mathscr{C}_0 = \Theta \cap \mathscr{B}_0$.

Step 1. Let $k = k + 1$. Sample $\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{km}$ uniformly and independently from $\mathscr{C}_{k-1}$. Let $\mathscr{V}_k = \mathscr{V}_{k-1} \cup \{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{km}\}$. Determine $a_k(\mathbf{x})$ according to the SAR for every $\mathbf{x}$ in $\mathscr{V}_k$. For all $\mathbf{x} \in \mathscr{V}_k$, take $a_k(\mathbf{x})$ observations, and update $N_k(\mathbf{x})$ and $\bar{G}_k(\mathbf{x})$.

Step 2. Let $\hat{\mathbf{x}}_k^* = \arg\min_{\mathbf{x}\in\mathscr{V}_k} \bar{G}_k(\mathbf{x})$. For each i, $i = 1, 2, \ldots, d$, let $\bar{x}_k^{(i)} = \max\{x_{k1}^{(i)}, \ldots, x_{km}^{(i)}\}$ and $\underline{x}_k^{(i)} = \min\{x_{k1}^{(i)}, \ldots, x_{km}^{(i)}\}$. If $\bar{x}_k^{(i)} > \bar{b}_{k-1}^{(i)} - \Delta^{(i)}$, then let $\bar{b}_k^{(i)} = \bar{x}_k^{(i)} + \Delta^{(i)}$; otherwise, let $\bar{b}_k^{(i)} = \bar{b}_{k-1}^{(i)}$. If $\underline{x}_k^{(i)} < \underline{b}_{k-1}^{(i)} + \Delta^{(i)}$, then let $\underline{b}_k^{(i)} = \underline{x}_k^{(i)} - \Delta^{(i)}$; otherwise, let $\underline{b}_k^{(i)} = \underline{b}_{k-1}^{(i)}$. Let $\mathscr{B}_k = \prod_{i=1}^{d}[\underline{b}_k^{(i)}, \bar{b}_k^{(i)}]$ and $\mathscr{C}_k = \{\mathbf{x}\colon \mathbf{x} \in \Theta \cap \mathscr{B}_k \text{ and } \|\mathbf{x} - \hat{\mathbf{x}}_k^*\| \leqslant \|\mathbf{x} - \mathbf{y}\| \ \forall \mathbf{y} \in \mathscr{V}_k \text{ and } \mathbf{y} \neq \hat{\mathbf{x}}_k^*\}$, and go to Step 1.

To prove the convergence of Algorithm 2, we make the following assumption on the SAR. For a more general discussion on this type of assumption, see Andradóttir (2004).

Assumption 5. *The SAR guarantees that there exists a sequence* $\{r_0, r_1, \ldots\}$ *such that* $r_0 \geqslant 1$, $r_{k+1} \geqslant r_k$ *for all* $k \geqslant 0$, $r_k \to \infty$ *as* $k \to \infty$, $\min_{\mathbf{x}\in\mathscr{V}_k} N_k(\mathbf{x}) \geqslant r_k$ *and* $\lim_{k\to\infty} k^{d+1}\lambda(r_k, \epsilon) = 0$ *for any* $\epsilon < \epsilon^*$, *where* ϵ^* *is defined in Assumption* 3.

For any infinite sequence $\{\mathscr{V}_0, \mathscr{V}_1, \ldots\}$ generated by Algorithm 2, let $\mathscr{V}_\infty = \bigcup_{k=0}^{\infty} \mathscr{V}_k$. To establish local convergence for Algorithm 2 we need the following lemma. Lemma 2 is proved in the appendix.

Lemma 2. *The infinite sequence* $\{\hat{\mathbf{x}}_0^*, \hat{\mathbf{x}}_1^*, \ldots\}$ *generated by Algorithm* 2 *satisfies*

$$\mathrm{P}\{\mathscr{N}(\mathbf{y}) \subset \mathscr{V}_\infty \mid \hat{\mathbf{x}}_k^* = \mathbf{y} \ i.o.\} = 1.$$

Let $\mathscr{M}$ denote the set of local minimizers in Θ where $|\mathscr{M}|$ can be infinity. We have the following theorem.

Theorem 2. *If Assumptions* 3–5 *are satisfied, then the infinite sequence* $\{\hat{\mathbf{x}}_0^*, \hat{\mathbf{x}}_1^*, \ldots\}$ *generated by Algorithm* 2 *converges with probability* 1 *to the set* $\mathscr{M}$ *in the sense that* $\mathrm{P}\{\hat{\mathbf{x}}_k^* \notin \mathscr{M} \ i.o.\} = 0$.

Proof. We first show that

$$\mathrm{P}\left\{\lim_{k\to\infty} g(\hat{\mathbf{x}}_k^*) = \min_{\mathbf{x}\in\mathscr{V}_\infty} g(\mathbf{x})\right\} = 1, \tag{11}$$

which is equivalent to showing for any $\epsilon > 0$,

$$\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_k} g(\mathbf{x})\right| \geqslant \epsilon \ i.o.\right\} = 0.$$

Because $r_k \to \infty$ as $k \to \infty$ and $r_{k+1} \geqslant r_k$ for all $k > 0$ by Assumption 5, there exists K such that $r_k \geqslant r^*$ when $k \geqslant K$ for any sequence $\{\mathscr{V}_0, \mathscr{V}_1, \ldots\}$, where r^* is defined in Assumption 3. For $k \geqslant K$ and $\epsilon < 2\epsilon^*$, where ϵ^* is defined in Assumption 3,

$$\begin{aligned}
&\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_k} g(\mathbf{x})\right| \geqslant \epsilon\right\} \\
&\leqslant \mathrm{P}\left\{|g(\hat{\mathbf{x}}_k^*) - \bar{G}_k(\hat{\mathbf{x}}_k^*)| \geqslant \frac{\epsilon}{2}\right\} + \mathrm{P}\left\{\left|\bar{G}_k(\hat{\mathbf{x}}_k^*) - \min_{\mathbf{x}\in\mathscr{V}_k} g(\mathbf{x})\right| \geqslant \frac{\epsilon}{2}\right\} \\
&\leqslant \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \text{ for some } \mathbf{x} \in \mathscr{V}_k\right\} \\
&\quad + \mathrm{P}\left\{\left|\min_{\mathbf{x}\in\mathscr{V}_k} \bar{G}_k(\mathbf{x}) - \min_{\mathbf{x}\in\mathscr{V}_k} g(\mathbf{x})\right| \geqslant \frac{\epsilon}{2}\right\} \\
&\leqslant \mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \text{ for some } \mathbf{x} \in \mathscr{V}_k\right\} \\
&\quad + \mathrm{P}\left\{\max_{\mathbf{x}\in\mathscr{V}_k} |\bar{G}_k(\mathbf{x}) - g(\mathbf{x})| \geqslant \frac{\epsilon}{2}\right\} \quad \text{by Lemma 1} \\
&\leqslant 2\mathrm{P}\left\{|g(\mathbf{x}) - \bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \text{ for some } \mathbf{x} \in \mathscr{V}_k\right\}.
\end{aligned} \tag{12}$$

Let $B_k = \prod_{i=1}^{d}[\underline{b}_0^{(i)} - k\Delta^{(i)}, \bar{b}_0^{(i)} + k\Delta^{(i)}]$; then $\mathscr{B}_k \subset B_k$ for all k. Let $V_k = B_k \cap \mathscr{Z}^d$ be the set of d-dimensional integers in B_k; then $\mathscr{V}_k \subset V_k$ and $|V_k| \leqslant (b + \Delta k)^d$, where $b = \max_{1\leqslant i\leqslant d}\lceil \bar{b}_0^{(i)} - \underline{b}_0^{(i)} + 1\rceil$ and $\Delta = \max_{1\leqslant i\leqslant d}\lceil 2\Delta^{(i)}\rceil$. Now suppose that at each iteration k, we allocate r_k observations to all solutions in V_k, but we use COMPASS logic to

determine $\mathscr{V}_k$ and the number of observations allocated to solutions in $\mathscr{V}_k$. Then,

$$\begin{aligned}(12) &\leqslant 2\mathrm{P}\left\{|g(\mathbf{x})-\bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2} \text{ for some } \mathbf{x}\in V_k\right\}\\ &\leqslant 2|V_k|\sup_{\mathbf{x}\in V_k}\mathrm{P}\left\{|g(\mathbf{x})-\bar{G}_k(\mathbf{x})| \geqslant \frac{\epsilon}{2}\right\}\\ &\leqslant 2(b+\Delta k)^d\lambda(r_k,\epsilon/2) \quad \text{by Assumptions 3 and 5.}\end{aligned}$$

Then, by Assumption 5,

$$\begin{aligned}&\sum_{k=0}^{\infty}\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*)-\min_{\mathbf{x}\in\mathscr{V}_k}g(\mathbf{x})\right| \geqslant \epsilon\right\}\\ &\quad\leqslant K+\sum_{k=K}^{\infty}2(b+\Delta k)^d\lambda(r_k,\epsilon/2)<\infty.\end{aligned}$$

Thus, the first Borel-Cantelli lemma (Billingsley 1995) implies that

$$\mathrm{P}\left\{\left|g(\hat{\mathbf{x}}_k^*)-\min_{\mathbf{x}\in\mathscr{V}_k}g(\mathbf{x})\right| \geqslant \epsilon \ i.o.\right\}=0.$$

Therefore, we prove Equation (11).

By Assumption 4, we know that for any $\mathscr{V}_\infty$, $\min_{\mathbf{x}\in\mathscr{V}_\infty}g(\mathbf{x})\leqslant g(\mathbf{x}_0)\leqslant g(\mathbf{z})-\delta$ for any $\mathbf{z}\in\Pi^c\cap\Theta$. Therefore, $\mathrm{P}\{\hat{\mathbf{x}}_k^*\in\Pi^c \ i.o.\}=0$.

Now we prove that $\mathrm{P}\{\hat{\mathbf{x}}_k^*\notin\mathscr{M} \ i.o.\}=0$. Note that $\mathrm{P}\{\hat{\mathbf{x}}_k^*\notin\mathscr{M} \ i.o.\}\leqslant\mathrm{P}\{\hat{\mathbf{x}}_k^*\in\mathscr{M}^c \ i.o.\}=\mathrm{P}\{\hat{\mathbf{x}}_k^*\in\mathscr{M}^c\cap\Pi \ i.o.\}$, because $\mathrm{P}\{\hat{\mathbf{x}}_k^*\notin\Pi^c \ i.o.\}=0$. Because $|\mathscr{M}^c\cap\Pi|<\infty$, $\hat{\mathbf{x}}_k^*\in\mathscr{M}^c\cap\Pi \ i.o.$ implies there exists $\mathbf{y}\in\mathscr{M}^c\cap\Pi$ such that $\hat{\mathbf{x}}_k^*=\mathbf{y}$ infinitely often. Then, $\mathscr{N}(\mathbf{y})\subset\mathscr{V}_\infty$ with probability 1 by Lemma 2, which implies $\lim_{k\to\infty}g(\hat{\mathbf{x}}_k^*)\neq\min_{\mathbf{z}\in\mathscr{V}_\infty}g(\mathbf{z})$ with probability 1. Note that $\mathrm{P}(A)\leqslant\mathrm{P}(B)$ if A implies B with probability 1. Then, $\mathrm{P}\{\hat{\mathbf{x}}_k^*\in\mathscr{M}^c\cap\Pi \ i.o.\}\leqslant\mathrm{P}\{\lim_{k\to\infty}g(\hat{\mathbf{x}}_k^*)\neq\min_{\mathbf{z}\in\mathscr{V}_\infty}g(\mathbf{z})\}$. By Equation (11), we have $\mathrm{P}\{\hat{\mathbf{x}}_k^*\notin\mathscr{M} \ i.o.\}=0$. □

If $\lambda(r,\epsilon)$ is as in Equation (8), then the condition on r_k in Assumption 5 becomes

$$\lim_{k\to\infty}k^{d+1}/r_k=0. \tag{13}$$

One SAR that satisfies Equation (13) is $r_k=\beta k^{d+1+\gamma}$ for some positive constants β and γ.

Similarly, if $\lambda(r,\epsilon)$ is as in Equation (10), then the condition on r_k in Assumption 5 becomes

$$\lim_{k\to\infty}k^{d+1}e^{-r_kI(\epsilon/2)}=0, \tag{14}$$

where $I(\epsilon/2)=\inf_{\mathbf{x}\in\Theta}I(\mathbf{x},\epsilon/2)/2$. One SAR that satisfies Equation (14) is

$$r_k=\beta(\log(k))^{1+\gamma} \tag{15}$$

for some positive constants β and γ.

5. Construction and Sampling from the Most Promising Area

Algorithms 1 and 2 both employ uniform sampling from $\mathscr{C}_k$, the most promising area in each iteration. Sampling uniformly over the set of d-dimensional integers in an arbitrary d-dimensional real set is difficult, in general. In this section, we first provide a method for constructing $\mathscr{C}_k$ and then provide a scheme for sampling from it when Φ is a convex set in $\Re^d$. Note that when Φ is formed by a set of linear constraints, Φ is a convex set.

If there are two points $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\Re^d$, then the points that are equidistant from $\mathbf{x}_1$ and $\mathbf{x}_2$ lie on the plane that has the normal $\mathbf{x}_1-\mathbf{x}_2$ and passes through $(\mathbf{x}_1+\mathbf{x}_2)/2$. The equation of the plane is

$$(\mathbf{x}_1-\mathbf{x}_2)'\left(\mathbf{x}-\frac{\mathbf{x}_1+\mathbf{x}_2}{2}\right)=0. \tag{16}$$

For a given $\mathbf{x}\in\Re^d$, if the left-hand side of Equation (16) is greater than or equal to 0, then $\|\mathbf{x}-\mathbf{x}_1\|\leqslant\|\mathbf{x}-\mathbf{x}_2\|$; and vice versa.

At iteration k of Algorithms 1 and 2, we have a set of visited solutions $\mathscr{V}_k$ and the current sample best solution $\hat{\mathbf{x}}_k^*$. To unify the presentation, we let Ω denote a convex and compact set, $\Omega=\Phi$ in Algorithm 1 and $\Omega=\Phi\cap\mathscr{B}_k$ in Algorithm 2, where $\Theta=\Phi\cap\mathscr{Z}^d$ for both algorithms. Then,

$$\begin{aligned}\tilde{\mathscr{C}}_k=\Bigg\{\mathbf{x}\in\Omega\colon (\hat{\mathbf{x}}_k^*-\mathbf{y})'\left(\mathbf{x}-\frac{\hat{\mathbf{x}}_k^*+\mathbf{y}}{2}\right)\geqslant 0\\ \forall\,\mathbf{y}\in\mathscr{V}_k \text{ and } \mathbf{y}\neq\hat{\mathbf{x}}_k^*\Bigg\}\end{aligned} \tag{17}$$

is a compact and convex set and $\mathscr{C}_k=\tilde{\mathscr{C}}_k\cap\mathscr{Z}^d$. Therefore, we can use Algorithm Mix-D (Pichitlamken and Nelson 2003) to generate solutions in $\mathscr{C}_k$, which are asymptotically uniformly distributed.

Algorithm Mix-D is designed to sample the set of integer solutions $\mathscr{A}$ within a convex and compact set $\tilde{\mathscr{A}}$ in $\Re^d$. Mix-D starts with an integer point $\mathbf{x}_0$, samples an integer point $\mathbf{x}$ uniformly from a hyper-rectangle $\prod_{i=1}^d[l_i,u_i]$, which tightly covers $\tilde{\mathscr{A}}$, and then forms a line (or a direction) passing through $\mathbf{x}$ and $\mathbf{x}_0$. If Υ is the set of integer points that are on the line and also in the set $\mathscr{A}$, then $\mathbf{y}$ is sampled uniformly from Υ. Let $\mathbf{X}_1=\mathbf{y}$, so $\mathbf{X}_1$ is a sample generated by Algorithm Mix-D. Instead of running Mix-D one step to get a sample from $\mathscr{A}$, it is run for t steps. When t is large, $\mathbf{X}_t$ is approximately uniform in $\mathscr{A}$, because $\mathbf{X}_t$ becomes uniformly distributed in $\mathscr{A}$ as $t\to\infty$.

Algorithm Mix-D is essentially an acceptance-rejection type of sampling method, because $\mathbf{x}_0$ might be the only point in Υ. In that case, the Mix-D algorithm stays at the current point, which can be viewed as rejection of the direction. If the set $\tilde{\mathscr{A}}$ is skewed, the acceptance rate of a direction can be low, so Mix-D can keep sampling the same point for many steps before moving to a new point.

We designed a new algorithm to sample from $\mathscr{C}_k$, which is similar to Mix-D. We call it *revised Mix-D* (RMD). Instead of generating a direction by sampling uniformly within the covering hyper-rectangle, RMD uses coordinate directions. It chooses a direction uniformly from the set of coordinate directions, and then proceeds as in the Mix-D algorithm. The advantages of RMD are that the acceptance rate is typically higher and the algorithm is easier to implement; the disadvantage is that it converges to the uniform distribution more slowly. Because the uniformity of the sampling distribution is not required to prove the convergence of COMPASS algorithms, we use RMD.

Algorithm 3 (RMD)

Input: A compact convex region $\tilde{\mathscr{A}} \subset \Re^d$, a number of random points n to sample, a length of the warm-up period T, and a starting solution $\mathbf{x}_0 \in \mathscr{A}$, where $\mathscr{A} = \tilde{\mathscr{A}} \cap \mathscr{Z}^d$.

Output: n random integer points in $\mathscr{A}$.

Procedure:

Step 0. Set $\mathbf{X}_0 = \mathbf{x}_0$, $t = 0$, the set of sampled points $\mathscr{S} = \varnothing$, and the number of sampled points $k = 0$.

Step 1. Set $t = t + 1$. Uniformly sample an integer I from 1 to d. Let $\ell(\mathbf{X}_{t-1}, I)$ be the line passing through $\mathbf{X}_{t-1}$ and parallel to the x_I coordinate axis. Then, $\ell(\mathbf{X}_{t-1}, I)$ intersects with the boundary of $\tilde{\mathscr{A}}$ at two points $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\Upsilon(\mathbf{X}_{t-1}, I)$ denote the set of integer points on $\ell(\mathbf{X}_{t-1}, I)$ and between $\mathbf{c}_1$ and $\mathbf{c}_2$.

Step 2. Sample $\mathbf{X}_t$ uniformly from the set $\Upsilon(\mathbf{X}_{t-1}, I)$.

Step 3. If $t < T$, go to Step 1; otherwise, let $\mathscr{S} = \mathscr{S} \cup \{\mathbf{X}_T\}$ and $k = k + 1$. If $k = n$, return the set $\mathscr{S}$; otherwise, set $t = 0$, $\mathbf{X}_0 = \mathbf{X}_T$, and go to Step 1.

REMARK. It is worthwhile to note that finding an initial point $\mathbf{x}_0$ in $\mathscr{A}$ is usually an NP-hard problem. For COMPASS algorithms, however, no effort is required to find a starting point for RMD because $\hat{\mathbf{x}}_k^*$ serves as the starting point for sampling from $\mathscr{C}_k$.

The next theorem shows that the RMD algorithm has a limiting uniform distribution over $\mathscr{C}_k$ if every two integer points in $\mathscr{C}_k$ are accessible to each other via the Markov chain constructed by RMD. The theorem also shows that the RMD algorithm always satisfies Inequality (7) if the algorithm stops in finite time, even if some solutions in $\mathscr{C}_k$ are not accessible from $\mathbf{x}_0$, and thus guarantees the convergence of Algorithms 1 and 2.

THEOREM 3. *Let $\mathscr{H}$ be the set of integer points in $\mathscr{A}$ which are accessible from $\mathbf{x}_0$ by RMD. Then, $\{\mathbf{X}_t\colon t \geqslant 0\}$ generated by RMD forms an ergodic Markov chain over $\mathscr{H}$, and the limiting distribution of the Markov chain is uniform over $\mathscr{H}$. If the algorithm is run for T steps, then for all $\mathbf{x} \in \mathscr{N}(\mathbf{x}_0)$ there exists a constant $c > 0$ such that*

$$\mathrm{P}\{\mathbf{X}_T = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{x}_0\} \geqslant c. \tag{18}$$

PROOF. Clearly, $\{\mathbf{X}_t\colon t \geqslant 0\}$ has the Markov property. If $\mathbf{x}$ and $\mathbf{y}$ are both in $\mathscr{H}$, then $\mathbf{x}$ communicates with $\mathbf{y}$ because both $\mathbf{x}$ and $\mathbf{y}$ communicate with $\mathbf{x}_0$. If $\mathbf{z} \notin \mathscr{H}$, then clearly $\mathbf{z}$ is not accessible from any points in $\mathscr{H}$. Therefore, $\mathscr{H}$ is a single closed communicating class and $\{\mathbf{X}_t\colon t \geqslant 0\}$ is irreducible. Further, $\{\mathbf{X}_t\colon t \geqslant 0\}$ is positive recurrent because $|\mathscr{H}| \leqslant |\mathscr{A}| < \infty$. It is also aperiodic because $\mathrm{P}\{\mathbf{X}_1 = \mathbf{x}_0 \mid \mathbf{X}_0 = \mathbf{x}_0\} > 0$. Therefore, $\{\mathbf{X}_t\colon t \geqslant 0\}$ forms an ergodic Markov chain, and it has a limiting distribution.

Let $\mathbf{x}$ and $\mathbf{y}$ be any two points in $\mathscr{H}$. If the line formed by $\mathbf{x}$ and $\mathbf{y}$ is parallel to one of the coordinate axes, denoted by $i(\mathbf{x}, \mathbf{y})$, then $\mathrm{P}\{\mathbf{X}_1 = \mathbf{y} \mid \mathbf{X}_0 = \mathbf{x}\} = 1/[d|\Upsilon(\mathbf{x}, i(\mathbf{x}, \mathbf{y}))|]$. Note that $\Upsilon(\mathbf{x}, i(\mathbf{x}, \mathbf{y}))$ is the same as $\Upsilon(\mathbf{y}, i(\mathbf{y}, \mathbf{x}))$, so $\mathrm{P}\{\mathbf{X}_1 = \mathbf{y} \mid \mathbf{X}_0 = \mathbf{x}\} = \mathrm{P}\{\mathbf{X}_1 = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{y}\}$. Otherwise, $\mathrm{P}\{\mathbf{X}_1 = \mathbf{y} \mid \mathbf{X}_0 = \mathbf{x}\} = \mathrm{P}\{\mathbf{X}_1 = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{y}\} = 0$. Therefore, $\{\mathbf{X}_t\colon t \geqslant 0\}$ is a doubly stochastic ergodic Markov chain, and its limiting distribution is uniform over $\mathscr{H}$.

If $\mathbf{y}$ is accessible from $\mathbf{x}$ in one step, then

$$\mathrm{P}\{\mathbf{X}_1 = \mathbf{y} \mid \mathbf{X}_0 = \mathbf{x}\} = \frac{1}{d}\frac{1}{|\Upsilon(\mathbf{x}, I)|} \geqslant \frac{1}{d}\frac{1}{|\mathscr{A}|}.$$

Thus, for any $\mathbf{x} \in \mathscr{N}(\mathbf{x}_0)$,

$$\begin{aligned}
&\mathrm{P}\{\mathbf{X}_T = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{x}_0\} \\
&\quad \geqslant \mathrm{P}\{\mathbf{X}_T = \mathbf{x} \mid \mathbf{X}_{T-1} = \mathbf{x}_0\} \\
&\qquad \cdot \mathrm{P}\{\mathbf{X}_{T-1} = \mathbf{x}_0 \mid \mathbf{X}_{T-2} = \mathbf{x}_0\} \cdots \mathrm{P}\{\mathbf{X}_1 = \mathbf{x}_0 \mid \mathbf{X}_0 = \mathbf{x}_0\} \\
&\quad \geqslant \left(\frac{1}{d|\mathscr{A}|}\right)^T = c.
\end{aligned}$$

The last inequality follows from the fact that both $\mathbf{x}_0$ and $\mathbf{x}$ are accessible from $\mathbf{x}_0$ in one step. □

It is easy to conclude from Theorem 3 that if all points in $\mathscr{A}$ are accessible to each other ($\mathscr{H} = \mathscr{A}$), then the sampling distribution induced by the RMD algorithm is asymptotically uniform over $\mathscr{A}$.

6. Numerical Experiments

In this section, we present numerical results obtained by applying COMPASS algorithms to solve DOvS problems. We first study the properties of COMPASS, and we then apply COMPASS to an inventory problem and compare COMPASS to other widely used algorithms.

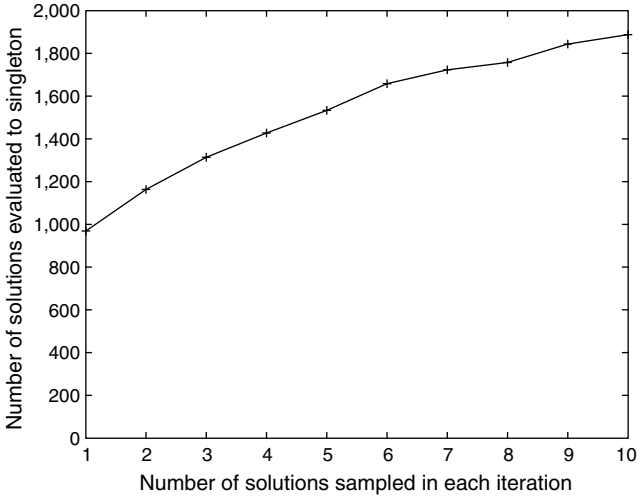
6.1. Local Convergence

Consider the following 10-dimensional quadratic function:

$$g(\mathbf{x}) = x_1^2 + x_2^2 + \cdots + x_{10}^2 + 1, \tag{19}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_{10})'$. The problem has only one local optimal solution, $\mathbf{x}^* = (0, 0, \ldots, 0)'$, with $g(\mathbf{x}^*) = 1$. We let $x_i \in [-100, 100] \cap \mathscr{Z}^{10}$ and let $\mathbf{x}_0 = (80, 80, \ldots, 80)'$. The problem has 1.08×10^{23} feasible solutions. To study local convergence of COMPASS, we let $G(\mathbf{x}, \psi) = g(\mathbf{x})$, which means the function can be evaluated without noise. Figure 1 shows the number of solutions

Figure 1. Number of solutions evaluated to singleton.



evaluated before COMPASS claims that the optimal solution is founded (i.e., the most promising area is a singleton) as a function of the number of new solutions sampled, m, on each iteration, averaged over 50 macroreplications.

From Figure 1 we see that, on average, fewer than 1,000 function evaluations are required not only to visit $\mathbf{x}^*$, but also to claim it is an optimal solution when the number of solutions sampled in each iteration is set to 1. Note that to claim that $\mathbf{x}^*$ is a *global* optimal solution, one has to evaluate all 1.08×10^{23} solutions. This illustrates the merit of local convergence.

Figure 1 also suggests that we want to set the number of solutions sampled in each iteration as small as possible. Unfortunately, this implies that a most promising area needs to be constructed for each solution sampled, which leads to a high computation cost per sample. Another reason to sample more than one solution in each iteration is that a problem may have multiple local optimal solutions. Sampling more than one solutions increases the chance to converge to a better local optimal solution. In the rest of this section, we set the number of solutions sampled in each iteration to 5.

6.2. Adaptiveness

Consider a classic (s, S) inventory problem (Koenig and Law 1985, Pichitlamken and Nelson 2003) in which the level of inventory of some discrete unit is periodically reviewed. The goal is to select s and S such that the steady-state expected inventory cost per review period is minimized. The constraints on s and S are $S - s \geqslant 10$, $20 \leqslant s \leqslant 80$, $40 \leqslant S \leqslant 100$, and $s, S \in \mathcal{Z}$. The optimal inventory policy is $(20, 53)$ with expected cost/period of 111.1265. To reduce the initial-condition bias, the average cost per period in each replication is computed after the first 100 review periods and averaged over the subsequent 30 periods.

We set the computational budget for this problem to 10,000 simulation observations and use the equal SAR with $N_k = \min\{5, \lceil 5(\log k)^{1.01} \rceil\}$. This SAR satisfies Equation (15) with $\beta = 5$ and $\gamma = 0.01$. We allocate at least five observations to each newly visited solution. The solutions visited through iterations 1, 5, 10 and the end of the COMPASS run are recorded and plotted. Figures 2 and 3 are two of the plots, with the optimal solution denoted as $\star$.

Figure 2 shows what happens in a typical run. The visited solutions concentrate quickly in the neighborhood of the optimal solution. Figure 3 is a somewhat rare case, where the visited solutions concentrate far away from the optimal solution in the beginning, then gradually shift to the neighborhood of the optimal solution. The reason the visited solutions initially concentrate in the wrong area is because of the noise in the simulation outputs, which makes some inferior solution appear to be a local optimal solution. Because COMPASS keeps allocating simulation observations to all visited solutions, it eventually escapes from the neighborhood of a nonoptimal solution and moves toward the optimal solution.

6.3. Constrained vs. Partially Constrained or Unconstrained

Because many partially constrained or unconstrained problems can be solved as a fully constrained problem by adding a set of boundary constraints to the feasible region, one can argue that there is no need to develop an algorithm for solving partially constrained or unconstrained problems directly. In this subsection, we study the effect of adding boundary constraints to partially constrained or unconstrained DOvS problems, relative to using Algorithm 2.

We use Problem (19) with $G(\mathbf{x}) = g(\mathbf{x}) + \epsilon(\mathbf{x})$, letting $\epsilon(\mathbf{x})$ be normally distributed with mean 0 and standard deviation $0.1g(\mathbf{x})$. We apply Algorithm 1 to the problem by adding boundary constraints $-500 \leqslant x_i \leqslant 500$, $i = 1, 2, \ldots, 10$, and also apply Algorithm 2 to the unconstrained problem. Both algorithms use the same starting point $(80, 80, \ldots, 80)'$. Results are shown in Figure 4, where the solid line is the average sample path over 50 macroreplications from Algorithm 2 and the dashed line is the average sample path over 50 macroreplications from Algorithm 1.

We can see that Algorithm 2 works better than Algorithm 1. Therefore, we conclude that adding arbitrary boundary constraints to a DOvS problem that is naturally partially constrained or unconstrained and solving it with an algorithm for fully constrained problems is not always an efficient approach, especially when the boundaries are set excessively large.

6.4. An Illustrative Example

Assemble-to-order systems have been studied recently in the supply chain management literature (Glasserman and Wang 1998, Chen et al. 2002, Iravani et al. 2003). The

Figure 2. Solutions visited by COMPASS in a typical run.

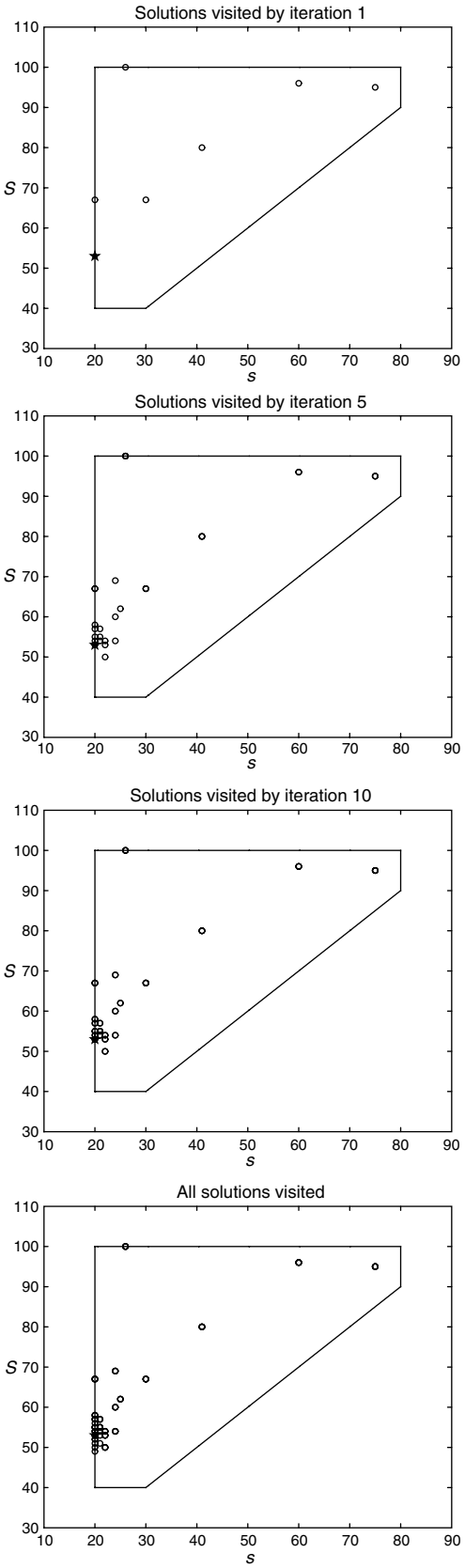


Figure 3. Solutions visited by COMPASS in an atypical run.

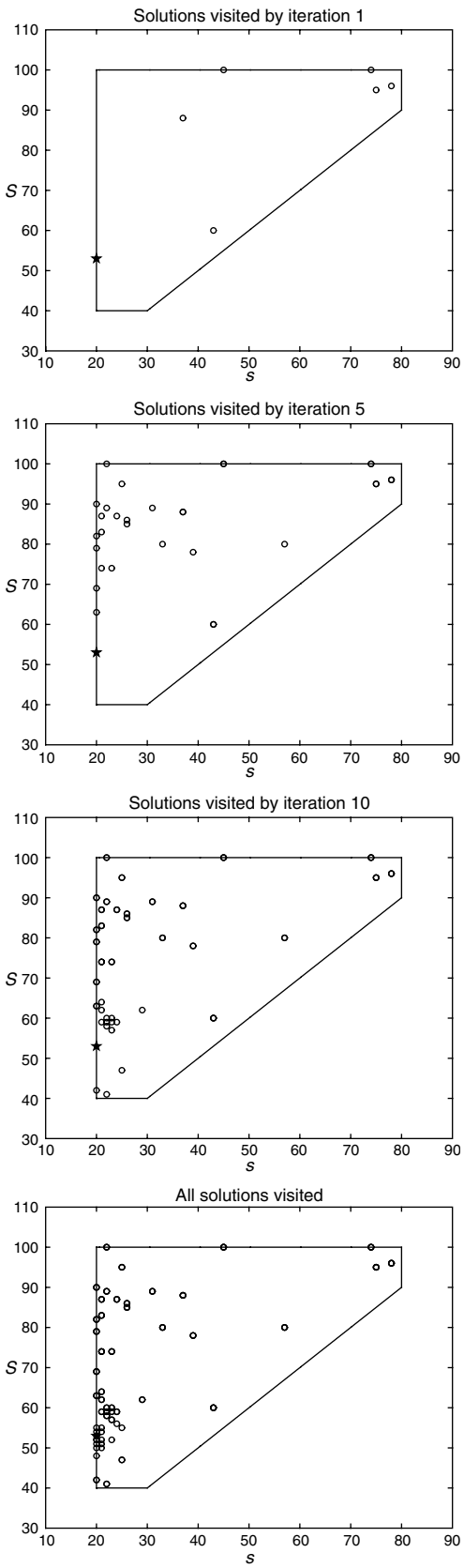
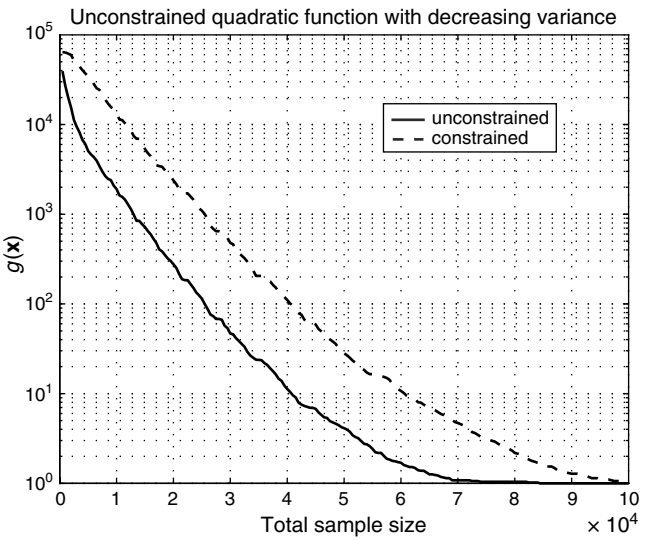


Figure 4. Unconstrained vs. constrained COMPASS performance.



models considered have the following general features: Items are made to stock to supply variable demands for finished products, and multiple finished products are assembled to order from the items. The system operates using a continuous-review base-stock policy under which each demand for a unit of an item triggers a replenishment order for that item. Items are produced one at a time on dedicated facilities, and production intervals are often stochastic.

In this section we study the following assemble-to-order system. The system has eight items, $I_1, I_2, \ldots, I_8$, and five types of customers, $T_1, T_2, \ldots, T_5$. Different types of customers come into the system as Poisson arrival processes with different rates, $\lambda_1, \lambda_2, \ldots, \lambda_5$, and each of them requires a set of key items and a set of nonkey items. If any of the key items are out of stock, the customer leaves. If all key items are in stock, the customer buys the product assembled from all the key items and the available nonkey items. Each item sold brings a profit, p_i, $i = 1, 2, \ldots, 8$, and each item in inventory has a holding cost per period, h_i, $i = 1, 2, \ldots, 8$. There are inventory capacities for each item, $C_1, C_2, \ldots, C_8$, such that $1 \leqslant x_i \leqslant C_i$, and the production time for each item is normally distributed with mean μ_i and variance σ_i^2, $i = 1, 2, \ldots, 8$, truncated at 0. All parameters used are included in Tables 1 and 2. We are interested in finding the optimal inventory level for each item to maximize the expected total profit per period.

Note that this problem cannot be solved by the algorithms in the literature because the production times are not exponentially distributed. When the production times are exponentially distributed, Iravani et al. (2003) provide a numerical approach to evaluate the objective function when the dimension of the problem is low. However, when the dimension of the problem is high, as in our example, the approach becomes extremely slow. When COMPASS is used to solve this problem, the production times can be from any distributions, and the dimension of the problem can be high.

Table 1. Parameters related to items.

Item	p_i	h_i	μ_i	σ_i	C_i
1	1	2	0.15	0.0225	20
2	2	2	0.40	0.0600	20
3	3	2	0.25	0.0375	20
4	4	2	0.15	0.0225	20
5	5	2	0.25	0.0375	20
6	6	2	0.08	0.0120	20
7	7	2	0.13	0.0195	20
8	8	2	0.40	0.0600	20

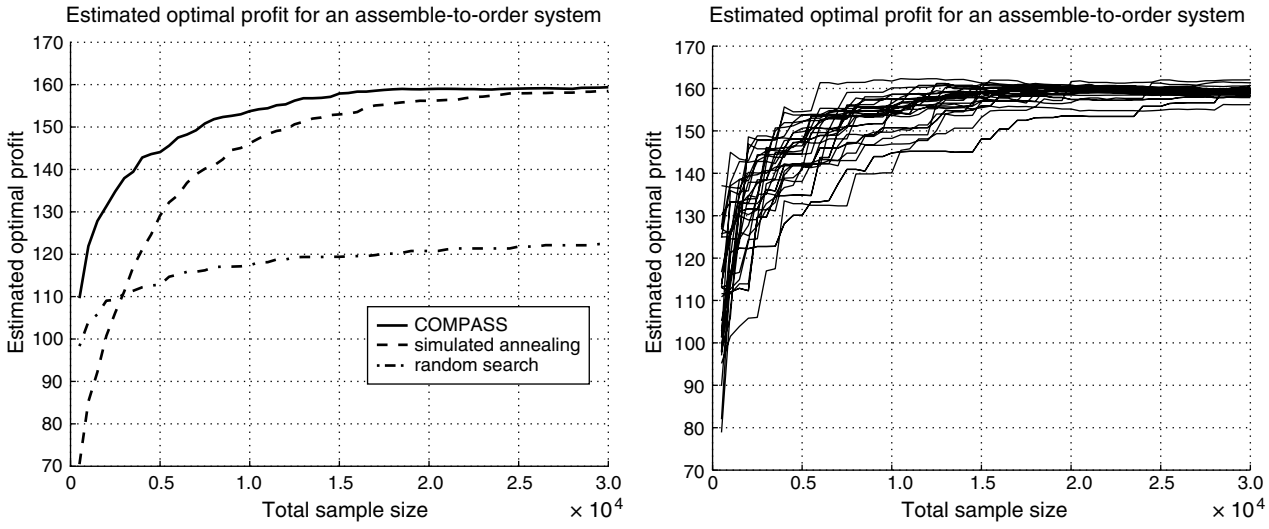
For the simulation we set a warm-up period of 20 periods and then averaged profit over the next 50 periods. We applied Algorithm 1 to solve this problem with randomly generated starting solutions. Because the true function values are not available, we compare the estimated profits of the sample best solutions from COMPASS, pure random search and simulated annealing. In the pure random search, we sample solutions uniformly from the feasible region using the RMD algorithm and apply the same simulation-allocation rule used in Algorithm 1 to allocate simulation experiments. The simulated annealing algorithm we use from Alrefaei and Andradóttir (1999) has a constant temperature and is not sensitive to the setting of the temperature. We set the temperature to 1. We also let the search neighborhood of $\mathbf{x}$ be its local neighborhood $\mathcal{N}(\mathbf{x})$ and set the number of observations taken from each visit of a solution to 5. Note that the performance of simulated annealing might be improved by careful selection of the neighborhood structure. However, the most appropriate structure is usually problem dependent and hard to determine, so the local neighborhood is often used in practice. For both COMPASS and pure random search, we set the sample size for each visited solution through iteration k as $\min\{5, \lceil 5(\log k)^{1.01} \rceil\}$.

The lines in the left panel of Figure 5 are the sample paths of COMPASS, simulated annealing and random search averaged over 30 macroreplications of the algorithms. We see that COMPASS works better than both simulated annealing and random search. The right panel of Figure 5 shows all 30 sample paths of COMPASS; all of

Table 2. Parameters related to customers.

		Key item						Nonkey	
Customer	λ_j	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
1	3.6	1	0	0	1	0	1	1	0
2	3.0	1	0	0	0	1	1	1	0
3	2.4	0	1	0	1	0	1	0	0
4	1.8	0	0	1	1	0	1	0	1
5	1.2	0	0	1	0	1	1	1	0

Figure 5. Assemble-to-order system optimization.

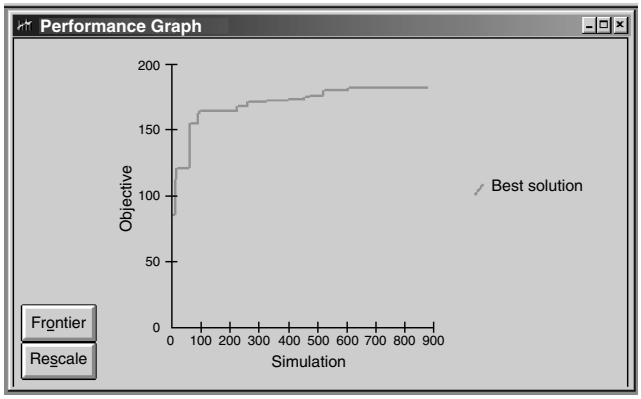


them converge to similar profits within 30,000 simulation observations.

We also ran OptQuest for Arena[2] on this problem, testing a variety of different customizable settings for that software. Directly comparing the performance of COMPASS to a commercial product like OptQuest is difficult, because the goals of the two procedures are somewhat different: COMPASS is designed to, and provably does, converge to a local optimal solution, while OptQuest attempts to find the very best solution it can within the time given by the analyst, without any convergence guarantee. Nevertheless, some useful general comments can be made that illustrate the strengths and areas of improvement for COMPASS.

Progress of the Search. Initially, OptQuest makes more rapid progress toward good solutions than COMPASS does, as one would expect because OptQuest uses smart local-improvement schemes for exactly this purpose. Figure 6 shows an OptQuest performance graph when the default settings are used; the x-axis shows the number of solutions visited, and it is clear that OptQuest's initial trajectory is quite steep.

Figure 6. OptQuest performance for the assemble-to-order problem.



Error Control. As the search progresses, COMPASS refines the estimated performance of every solution visited during the search. OptQuest, on the other hand, allocates a fixed number of replications (by default, 1), or a number of replications within a user-specified range, to each solution it visits. Thus, COMPASS might waste effort on solutions that are not competitive, but in the end it will produce a good estimate of the performance of the locally optimal solution it identifies. Although OptQuest will not waste replications on inferior solutions, it will retain whatever error was present in the initial estimate of each solution's performance, and this can be large or small depending on the variability of the simulation response. In the assemble-to-order problem, OptQuest typically found good solutions, but it had poor estimates of their performance if the number of replications was chosen too small. The performance graph in Figure 6 shows that the estimated expected profit is far higher than *any* solution in the feasible region actually attains.

Ending the Search. As COMPASS closes in on a local optimal solution, it stops searching for new solutions while continuing to refine the estimated performance of the solutions it has visited in a way that guarantees convergence. By design, OptQuest searches for better and better solutions, which could require exhausting the feasible space unless an automatic stopping rule is invoked (terminate the search when an improved solution has not been found in $50 +$ (number of decision variables)^2 trials). In our experiments OptQuest visited many more feasible solutions than COMPASS did, even when automatic stopping was turned on.

This comparison points out several ways COMPASS could be improved, the two most obvious being (a) to incorporate smarter simulation-allocation rules to reduce

the sampling from (apparently) inferior solutions, and (b) to incorporate local-improvement schemes to enhance performance early in the search. Fortunately, COMPASS provides an open framework within which such features can be added without sacrificing provable convergence.

7. Conclusions

In this paper, we present a new algorithm, COMPASS, which can solve both constrained and partially constrained or unconstrained DOvS problems with integer-ordered decision variables for both terminating and steady-state simulation. We show that, with probability 1, COMPASS converges to the set of local optimal solutions under mild conditions. We also show that COMPASS solves a wide range of problems, including problems with very large numbers of feasible solutions, and works better than some of the existing algorithms.

To further improve the performance of COMPASS, we are investigating smart simulation-allocation rules that take advantage of sample information obtained in previous iterations, and incorporating local-improvement schemes to enhance performance early in the search. Investigation of the behavior and performance of COMPASS in higher dimensions is another subject of future research.

Appendix

Proof of Lemma 1. Note that because

$$\min_{i=1,\ldots,n} b_i + \min_{i=1,\ldots,n} (a_i - b_i) \leqslant \min_{i=1,\ldots,n} (b_i + a_i - b_i) \leqslant \min_{i=1,\ldots,n} b_i + \max_{i=1,\ldots,n} (a_i - b_i),$$

we have

$$\min_{i=1,\ldots,n} (a_i - b_i) \leqslant \min_{i=1,\ldots,n} a_i - \min_{i=1,\ldots,n} b_i \leqslant \max_{i=1,\ldots,n} (a_i - b_i).$$

Therefore,

$$\left| \min_{i=1,\ldots,n} a_i - \min_{i=1,\ldots,n} b_i \right| \leqslant \max_{i=1,\ldots,n} |a_i - b_i|. \quad \square$$

Proof of Lemma 2. Let $\mathscr{D}_k = \{\mathbf{x}\colon \mathbf{x} \in \mathscr{Z}^d \text{ and } \|\mathbf{x} - \hat{\mathbf{x}}_k^*\| \leqslant \|\mathbf{x} - \mathbf{z}\| \ \forall \mathbf{z} \in \mathscr{V}_k \text{ and } \mathbf{z} \neq \mathbf{x}_0\}$, the set of integer points that are at least as close to $\hat{\mathbf{x}}_k^*$ as to other solutions in $\mathscr{V}_k$. Then, $\mathscr{C}_k = \mathscr{D}_k \cap \mathscr{B}_k \cap \Theta$.
Let

$$\mathscr{D}_0^{\mathbf{y}} = \begin{cases} \mathscr{D}_0 & \text{if } \hat{\mathbf{x}}_0^* = \mathbf{y}, \\ \mathscr{Z}^d & \text{if } \hat{\mathbf{x}}_0^* \neq \mathbf{y}, \end{cases}$$

and for $k \geqslant 1$,

$$\mathscr{D}_k^{\mathbf{y}} = \begin{cases} \mathscr{D}_k & \text{if } \hat{\mathbf{x}}_k^* = \mathbf{y}, \\ \mathscr{D}_{k-1} & \text{if } \hat{\mathbf{x}}_k^* \neq \mathbf{y}. \end{cases}$$

Then, $\mathscr{D}_{k+1}^{\mathbf{y}} \subset \mathscr{D}_k^{\mathbf{y}}$ for $k = 0, 1, \ldots$. Note that if $\hat{\mathbf{x}}_k^* = \mathbf{y}$, then all new solutions sampled at iteration $k+1$ are sampled from $\mathscr{D}_k^{\mathbf{y}}$ because $\mathscr{C}_k \subset \mathscr{D}_k = \mathscr{D}_k^{\mathbf{y}}$. For any $\mathbf{x} \in \mathscr{N}(\mathbf{y})$, if $\mathbf{x} \notin \mathscr{V}_k$, then $\mathbf{x} \in \mathscr{D}_k^{\mathbf{y}}$ because $\|\mathbf{x} - \mathbf{y}\| = 1$. If $|\mathscr{D}_K^{\mathbf{y}}| < \infty$ for some $K > 0$, then for any $k \geqslant K$,

$$\mathrm{P}\{\mathbf{x} \in \mathscr{V}_{k+1} \mid \hat{\mathbf{x}}_k^* = \mathbf{y}\} \geqslant \frac{1}{|\mathscr{D}_K^{\mathbf{y}}|}.$$

Because $\hat{\mathbf{x}}_k^* = \mathbf{y}$ infinitely often, then $\mathbf{x} \in \mathscr{V}_\infty$ with probability 1. Therefore, it suffices to prove $\mathrm{P}\{|\mathscr{D}_k| = \infty \ i.o.\} = 0$.

To show this, we build a new Cartesian coordinate system centered at $\mathbf{y}$ with one of its coordinate axis passing through $\mathbf{y} + [1, \sqrt{2}, \sqrt{2}, \ldots, \sqrt{2}]'$ in the original coordinate system. Then, the d-dimensional coordinate planes in the new coordinate system divide the space into 2^d d-dimensional compartments. See Figure 7 for a two-dimensional example.

First, we claim that there are no integer solutions (in the original coordinate system), other than $\mathbf{y}$, on the axes of the new coordinate systems. Because of symmetry, any integer solution on any coordinate axis of the new coordinate system has a corresponding integer solution on the axis passing through $\mathbf{y}$ and $\mathbf{y} + [1, \sqrt{2}, \sqrt{2}, \ldots, \sqrt{2}]'$. Suppose that there is an integer solution on the axis $\mathbf{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(d)}]$ (in the original coordinate system). Then, $(x^{(2)} - y^{(2)})/(x^{(1)} - y^{(1)}) = \cdots = (x^{(d)} - y^{(d)})/(x^{(1)} - y^{(1)}) = \sqrt{2}$. However, $\sqrt{2}$ is an irrational number and it cannot be represented by p/q where p and q are both integers. Therefore, there are no integer solutions other than $\mathbf{y}$ on the new coordinate axes.

For every integer solution $\mathbf{x}$, $\mathbf{x} \neq \mathbf{y}$, in a d-dimensional compartment $\mathscr{Q}$, the plane that is perpendicular to, and passes through the midpoint of the line segment $\overline{\mathbf{x}, \mathbf{y}}$, cuts off the compartment in the sense that $|\{\mathbf{z} \in \mathscr{Q}\colon \|\mathbf{z} - \mathbf{y}\| \leqslant \|\mathbf{z} - \mathbf{x}\|\}| < \infty$. See Figure 8 for a two-dimensional example. In iteration k of Algorithm 2, if $\hat{\mathbf{x}}_k^* = \mathbf{y}$ and $\mathscr{V}_k$ includes

Figure 7. New Cartesian coordinate system in two-dimensional space.

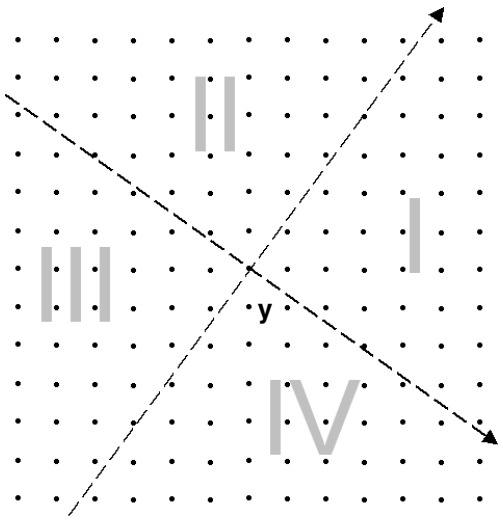
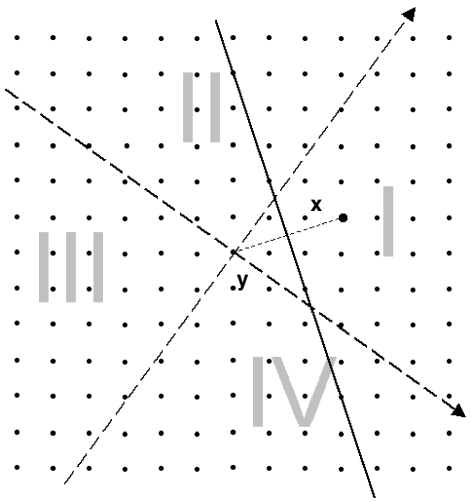


Figure 8. The line, formed by **x**, cuts off quadrant I.



solutions in all compartments, then $|\mathscr{D}_k^{\mathbf{y}}| < \infty$. See Figure 9 for a two-dimensional example. Therefore, it suffices to prove that all 2^d compartments are visited with probability 1.

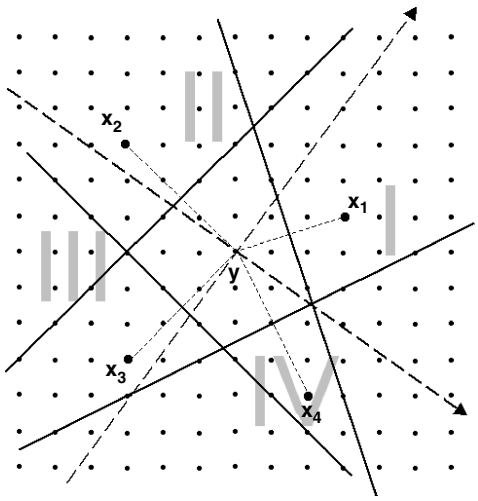
Note that when a compartment $\mathscr{Q}$ is visited infinitely many times, then $|\mathscr{D}_k^{\mathbf{y}} \cap \mathscr{Q} \backslash \{\mathbf{y}\}|$ goes to 0. Because the number of compartments is finite, no integer solutions are on the coordinate axes except **y**, and $\mathscr{D}_k^{\mathbf{y}} \cap \mathscr{Q} \backslash \{\mathbf{y}\} \neq \varnothing$ if compartment $\mathscr{Q}$ has never been visited, then every compartment is visited with probability 1.

Therefore, $\mathrm{P}\{|\mathscr{D}_k| = \infty \ i.o.\} = 0$ because, with probability 1, there exists a $K > 0$, that possibly depends on the sample path, such that $|\mathscr{D}_k| \leqslant |\mathscr{D}_K| < \infty$ for all $k \geqslant K$. □

Endnotes

1. OptQuest® is a registered trademark of OptTek Systems, Inc.

Figure 9. $|\mathscr{D}_k^{\mathbf{y}}| < \infty$ if there are solutions in each quadrant.



2. Arena® is a registered trademark of Rockwell Software, Inc.

Acknowledgments

The authors thank Sigrún Andradóttir of Georgia Institute of Technology and Tito Homem-de-Mello of Northwestern University for their ideas and assistance. This research was partially supported by National Science Foundation grant DMI-0217690 and by General Motors R&D.

References

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

Alrefaei, M. H., S. Andradóttir. 2001. A modification of the stochastic ruler method for discrete stochastic optimization. *Eur. J. Oper. Res.* **133** 160–182.

Andradóttir, S. 1995. A method for discrete stochastic optimization. *Management Sci.* **41** 1946–1961.

Andradóttir, S. 1996. A global search method for discrete stochastic optimization. *SIAM J. Optim.* **6** 513–530.

Andradóttir, S. 1998. Simulation optimization. J. Banks, ed. *The Handbook of Simulation*. Wiley, New York, 307–334.

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

Andradóttir, S. 2004. Simulation optimization with countably infinite feasible regions: Efficiency and convergence. Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

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

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

Chen, C. H., J. Lin, E. Yücesan, S. E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic System: Theory Applications* **10** 251–270.

Chen, F., E. Markus, G. Lin, D. Yao. 2002. Inventory-service optimization in configure-to-order systems. *Manufacturing Service Oper. Management* **4** 114–132.

Dembo, A., O. Zeitouni. 1998. *Large Deviations Techniques and Applications*, 2nd ed. Springer-Verlag, New York.

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

Gerencsér, L., S. D. Hill, Z. Vágó. 1999. Optimization over discrete sets via SPSA. *Proc. 38th Conf. Decision and Control*. 1791–1795.

Glasserman, P., Y. Wang. 1998. Leadtime-inventory trade-offs in assemble-to-order systems. *Oper. Res.* **46** 858–871.

Glover, F., J. P. Kelly, M. Laguna. 1996. New advances and applications of combining simulation and optimization. *Proc. 1996 Winter Simulation Conf.* 144–152.

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

Ho, Y. C., C. G. Cassandras, C. H. Chen, L. Y. Dai. 2000. Ordinal optimization and simulation. *J. Oper. Res. Soc.* **51** 490–500.

Homem-de-Mello, T. 2000. Monto Carlo methods for discrete stochastic optimization. S. Uryasev, P. M. Pardalos, eds. *Stochastic Optimization: Algorithms and Applications*. Kluwer Academic Publishers, Boston, MA.

Hong, L. J., B. Nelson. 2004. Selecting the best system when the systems are revealed sequentially. Working paper, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Iravani, S. M. R., L. Luangkesorn, D. Simchi-Levi. 2003. On assemble-to-order systems with flexible customers. *IIE Trans.* **35** 389–403.

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

Koenig, L. W., A. M. Law. 1985. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Comm. in Statist.: Simulation and Comput.* **14** 719–734.

Ólafsson, S., J. Kim. 2002. Simulation optimization. *Proc. 2002 Winter Simulation Conf.* 79–84.

Pichitlamken, J., B. L. Nelson. 2003. A combined procedure for optimization via simulation. *ACM Trans. Modeling & Comput. Simulation* **13** 155–179.

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

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