Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

DISCRETE OPTIMIZATION VIA SIMULATION USING COORDINATE SEARCH

L. Jeff Hong

Department of Industrial Engineering and Logistics Management
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, CHINA

ABSTRACT

In this paper we propose a coordinate search algorithm to solve the optimization-via-simulation problems with integer-ordered decision variables. We show that the sequence of solutions generated by the algorithm converges to the set of local optimal solutions with probability 1 and the estimated optimal values satisfy a central limit theorem. We compare the coordinate search algorithm to the COMPASS algorithm proposed in Hong and Nelson (2004) through a set of numerical experiments. We see that the coordinate search has a better performance.

1 INTRODUCTION

Applications of optimization via simulation (OvS) have been increasing in areas including manufacturing (e.g., Vogt 2004), supply-chain management (e.g., Truong and Azadivar 2003), logistics (for instance, Wieland and Holden 2003), telecommunication (e.g., Baras 2003) and project management (e.g., April et al. 2004). There are at least two reasons for the burst of OvS applications: First, the objective of many simulation studies is to find the best system configuration. To find the best system configuration we often need to simulate a large number of system configurations, and advances in computer technology make it possible to simulate a large number of system configurations in a reasonable amount of time. Therefore, it becomes feasible to conduct an optimization process on a simulation model. Second, the advantages of OvS attract more modelers to use OvS to solve their optimization problems. To solve a practical optimization problem using a mathematical programming approach, a modeler first creates an equation-based model of the system and then optimizes the model using appropriate algorithms. This approach requires the modeler to have a good knowledge of optimization and make many (perhaps restrictive) approximations to ensure that the model is mathematically tractable. When using OvS, however, the modeler only needs to construct a simulation model and optimize it using OvS algorithms. Typically, simulation models are easier to construct, require fewer assumptions and may capture more details of the system than analytical models, especially when used to study complex systems. For a recent review on OvS research and practice, see Fu(2002).

In this paper we are interested in the OvS problems with integer-ordered decision variables, which may be called discrete OvS problems or DOvS problems. A number of methods have been proposed in the literature to solve DOvS problems, including sample average approximation method (Kelywegt et al. 2001), discrete simultaneous perturbation stochastic approximation method (Gerencsér et al. 1999) and random search algorithms, including stochastic ruler method (Yan and Mukai 1992), simulated annealing (Alrefaei and Andradóttir 1999), stochastic comparison method (Gong et al. 1999), nested partitions (Shi and Ólafsson 2000) and COMPASS algorithm (Hong and Nelson 2004). Among the random search algorithms mentioned above, COMPASS is locally convergent, i.e., converges to the set of local optimal solutions, and all other algorithms are globally convergent, i.e., converge to the set of globally optimal solutions.

To guarantee global convergence, the globally convergent algorithms need to simulate all feasible solutions infinitely often in the limit. When the problem has a large number of feasible solutions, simulating all feasible solutions is often not possible. To solve this problem Hong and Nelson (2004) suggest to focus on local convergence. To ensure local convergence (see Section 2 for the definition of local convergence), only the local optimal solutions and their neighboring solutions need to be simulated infinitely often in the limit. Therefore, locally convergent algorithms in general converge faster than globally convergent algorithm. Moreover, since not all feasible solutions are required to be simulated, COMPASS may also be applied to solve partially constrained or unconstrained DOvS problems that have an infinite number of feasible solutions.

In the paper we propose a coordinate search algorithm to solve both fully constrained DOvS problems and partially constrained or unconstrained DOvS problems. Coordinate search algorithm works as follows: In each iteration the algorithm first selects a coordinate direction. It then conducts a line search along the coordinate direction for a sample mean function and finds a local optimal solution. We are able to show that the sequence of solutions generated by the coordinate search algorithm converges with probability 1 to the set of local optimal solutions for both fully constraind DOvS problems and partially constrained or unconstrained DOvS problems. We also show that, under certain conditions, the estimate objective values generated by the algorithm satisfy a central limit theorem.

One important feature of the coordinate search algorithm is that it is *not* a random search algorithm. Random search algorithms, e.g., COMPASS, visit and simulate solutions randomly to find better solutions. When the probability of finding better solutions are small, random search algorithms do not work well. Coordinate search algorithm, on the other hand, uses a more systematic approach to search for better solutions. The numerical results show that the coordinate search algorithm has better performance than COMPASS and it may solve problems with higher dimensions compared to COMPASS.

The paper is orgnized as follow: The statement of DOvS problems and the definition of local minimizers are provided in Section 2, followed by an introduction to the coordinate search algorithm in Section 3. The asymptotic properties of the algorithms are discussed in Section 4, followed by the results from a comprehensive numerical studies in Section 5.

2 PROBLEM STATEMENT

We are interested in solving the following problem:

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

where $\Theta = \Omega \cap \mathcal{Z}^d$, Ω is a closed and convex 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. When Ω is bounded, Θ has a finite number of feasible solutions. Then Problem (1) is called a fully constrained problem. When Ω is unbounded, Θ has an infinite number of feasible solutions. Then Problem (1) is called a partially constrained or unconstrained problem. 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 $g(\mathbf{x}) = \mathrm{E}_\psi\left[G(\mathbf{x},\psi)\right]$ 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 $\mathbf{x}$.

Let $\mathcal{N}(\mathbf{x}) = \{\mathbf{y} : \mathbf{y}\in\Theta \text{ and } \|\mathbf{x}-\mathbf{y}\| = 1\}$ be the *neighborhood* of $\mathbf{x}\in\Theta$, where $\|\mathbf{x}-\mathbf{y}\|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. Define $\mathbf{x}\in\Theta$ to be a *local minimizer* of Problem (1) if $g(\mathbf{x}) \le g(\mathbf{y})$ for all $\mathbf{y}\in\mathcal{N}(\mathbf{x})$ or $\mathcal{N}(\mathbf{x}) = \emptyset$. The definition is consistent with Hong and Nelson (2004). Let $\mathcal{M}$ denote the set of local minimizers of the function g in Θ; locally convergent algorithms guarantee to converge to $\mathcal{M}$.

Let $G_i(\mathbf{x})$ denote the ith obserations taken from $\mathbf{x}$ and let $\bar{G}(\mathbf{x},r) = \sum_{i=1}^r G_i(\mathbf{x})/r$. We make the following assumption on $\bar{G}(\mathbf{x},r)$:

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

$$\mathrm{P}\left[\lim_{r\to\infty} \bar{G}(\mathbf{x},r) = g(\mathbf{x})\right] = 1.$$

Assumption 1 implies that the sample mean of $G(\mathbf{x},\psi)$ is an appropriate estimator of $g(\mathbf{x})$. Assumption 1 holds automatically for terminating simulations, where the simulation output is a sequence of i.i.d. random observations, according to the Strong Law of Large Numbers. Assumption 1 also holds for output from typical steady-state simulations.

Let $\mathbf{x}_0$ be the user provided starting solution. We require $\mathbf{x}_0$ to be a feasible solution. Moreover, $\mathbf{x}_0$ needs to satisfy the following assumption.

Assumption 2 *If Problem (1) is partially constrained or unconstrained, there exists a positive constant δ such that the level set $\mathcal{L} = \{\mathbf{x}\in\Theta : g(\mathbf{x}) \le g(\mathbf{x}_0)+\delta\}$ is finite.*

Assumption 2 is equivalent to Assumption 4 of Hong and Nelson (2004). However, Assumption 2 is in a form that is more common used in the optimization literature. Assumption 2 generally holds since simulation study typically has a benchmark system configuration, which could be the current configuration. Solutions that are far away from the benchmark configuration are typically inferior to the benchmark.

3 COORDINATE SEARCH ALGORITHM

If a solution is a local minimizer of Problem (1) it is also a local minimizer along all coordinate directions. The basic idea of the coordinate search is to conduct search along each direction sequentially until it converges. In the remainder of this paper we use x_i direction to denote the direction (without sign) along the ith coordinate axis. The algorithm starts with an initial solution $\mathbf{x}_0$ and sets $\hat{\mathbf{x}}_0^* = \mathbf{x}_0$, where $\hat{\mathbf{x}}_k^*$ denotes the sample best solution at the end of iteration k. In the first iteration, the algorithm conducts a line search along x_1 directions and finds $\hat{\mathbf{x}}_1^*$ which is sample local minimizer along the x_1 direction. Then in the second iteration it conducts a line search along x_2 directions and finds $\hat{\mathbf{x}}_2^*$ and so on so forth. After finishing all d coordinate directions it goes back to the x_1 direction. Notice that if there is no noise in the function evaluations it is easy to show that the algorithm converges, since the algorithm always moves to a better solution if the current best solution is not a local minimizer. We are also able to show that the same convergence property holds with probability 1 even if when the objective function values can only be estimated with noise.

The following is the coordinate search algorithm:

Algorithm 1: Coordinate Search

Step 0 : Find a starting point $\mathbf{x}_0 \in \Theta$. Set iteration count $k = 0$ and dimension count $i = 0$. Let $\hat{\mathbf{x}}_0^* = \mathbf{x}_0$.

Step 1 : Let $k = k+1$ and $i = i+1$. If $i > d$, where d is the dimension of Problem (1), let $i = 1$. Determine N_k, the number of simulation observations that will be allocated to a solution visited in iteration k. We require that $N_k \geq N_{k-1}$.

Step 2 : Find a local minimizer y^* to the following one-dimensional minimization problem:

$$\begin{array}{ll} \min & \bar{G}(\hat{\mathbf{x}}_{k-1}^* + y\mathbf{e}_i, N_k) \\ \text{subject to} & \hat{\mathbf{x}}_{k-1}^* + y\mathbf{e}_i \in \Theta \\ & y \in \mathcal{Z} \end{array} \quad (2)$$

where $\mathbf{e}_i$ is the ith column of the d-dimensional identity matrix. If Problem (1) is fully constrained, then let $\hat{\mathbf{x}}_k^* = \hat{\mathbf{x}}_{k-1}^* + y^*\mathbf{e}_i$. If Problem (1) is partially constrained or unconstrained, then let $\hat{\mathbf{x}}_k^* = \text{argmin}\{\bar{G}(\hat{\mathbf{x}}_{k-1}^* + y^*\mathbf{e}_i, N_k), \bar{G}(\mathbf{x}_0, N_k)\}$ to make sure that $\hat{\mathbf{x}}_k^*$ is not worse than $\mathbf{x}_0$. Then go to **Step 1**.

Remark: In **Step 1** one can choose any N_k such that $N_k \geq N_{k-1}$. To ensure the convergence of Algorithm 1 (see Theorems 1 and 2 in Section 4), we require that $N_k \to \infty$ as $k \to \infty$.

To reuse the simulation observations allocated before iteration k, we check every solution visited in iteration k to see if it has been visited through iteration $k-1$. If it has been visited and it has r_{k-1} observations, then we only allocate $N_k - r_{k-1}$ observations to the solution.

When N_k is fixed, the objective function of Problem (2) can be evaluated without noise. Therefore, Problem (2) is a one-dimensional *deterministic* discrete optimization problem. We may solve the problem by line search algorithms. We require the line search algorithm used in Algorithm 1 to satisfy the following conditions:

Condition 1 *The local minimizer y^* of Problem (2) satisfies $\bar{G}(\hat{\mathbf{x}}_{k-1}^* + y^*\ \mathbf{e}_i, N_k) \leq \bar{G}(\hat{\mathbf{x}}_{k-1}^*, N_k)$.*

Condition 2 *The number of solutions visited in the line search is bounded above by some constant $M > 0$ for all iterations.*

Condition 3 *If $y^* = 0$ is a local minimizer to Problem (2) and solutions in $\{\hat{\mathbf{x}}_{k-1}^* \pm \mathbf{e}_i\} \cap \mathcal{N}(\hat{\mathbf{x}}_{k-1})$ have all been visited through iteration $k-1$, then only $y = 0$ and $y = \pm 1$ can be evaluated at iteration k.*

Condition 1 requires that the sample best solution at the end of iteration k should be at least as good as the sample best solution at the end of iteration $k-1$ based on the N_k simulation observations allocated to each of them. This condition makes sure that the quality of the solution keeps improving.

Condition 2 ensures that only a finite number of solutions are visited in each iteration. If Problem (1) is fully constrained, Condition 2 is automatically satisfied since the number is bounded by $|\Theta|$. If Problem (1) is partially constrained or unconstrained, we need this condition to ensure the local convergence of Algorithm 1. However, we may set M to be a large number such that it almost has no impact on the algorithm.

Two approaches are typically used in the line search: backtracking and forward searching. In the backtracking line search algorithms, we first set the maximum distance along a descent direction and then move backwards until finding a local minimizer. In the forward line search algorithms, we move from the starting point gradually along a descent direction until finding a local minimizer. When backtracking line search is used, Condition 2 is automatically satisfied if the maximum distance is fixed for all iterations. When forward line search is used, however, we need to set a maximum distance to satisfy Condition 2.

If Algorithm 1 has converged to a solution and all solutions in the neighborhood of the solution are visited. Then in each iteration, we only need to compare the solution to its two neighboring solutions along x_i direction to check if the solution is still a local minimizer along the coordinate direction. If it is, then we find a local minimizer that satisfies Conditions 1 and 2; if not, we may conduct a full line search. This is the basic idea behind Condition 3. To guarantee the convergence of Algorithm 1, Condition 3 is not required. However, it is useful when analyzing the limiting distributions (see Section 4.2).

In the following of this section we provide a forward line search algorithm. The algorithm first evaluate $\hat{\mathbf{x}}_{k-1}^* + \mathbf{e}_i$ or $\hat{\mathbf{x}}_{k-1}^* - \mathbf{e}_i$ depending on their feasibilities and identify a descent direction, either $\mathbf{e}_i$ or $-\mathbf{e}_i$. Then the algorithm search along the decent direction. Initially, the step size of the search is large. It is shortened when an infeasible solution or a worse solution is visited. The algorithm will stop when a local minimizer is identified. As described in Algorithm 1, all visited solutions in the line search are checked to see if they have been visited before to reuse the simulation observations allocated before the iteration. To satisfy Condition 2, we set an upper bound for the maximum moving distance. To satisfy Condition 3, we first check if two neighboring solutions of $\hat{\mathbf{x}}_{k-1}^*$ have been visited. Notice that Condition 1 is satisfied automatically by the algorithm.

In the following description of the algorithm we assume that it is in iteration k of Algorithm 1 and the coordinate is x_i.

Algorithm 2: Line Search

Step 0 : If both $\hat{\mathbf{x}}^*_{k-1}+\mathbf{e}_i$ and $\hat{\mathbf{x}}^*_{k-1}-\mathbf{e}_i$ are not feasible (i.e., they are not in Θ), then stop and return $y^*=0$. Otherwise, let $y=1$ if $\hat{\mathbf{x}}^*_{k-1}+\mathbf{e}_i$ is feasible and $y=-1$ if it is not. Evaluate $\bar{G}(\hat{\mathbf{x}}^*_{k-1},N_k)$ and $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)$. If $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)<\bar{G}(\hat{\mathbf{x}}^*_{k-1},N_k)$, let $d=y$ and $z_0=1$; otherwise, let $d=-y$ and $z_0=0$. Let $y_0=d\cdot z_0$.

Step 1 : If $\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i+d\mathbf{e}_i$ has been visited before and $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i+d\mathbf{e}_i,N_k)\geq\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i,N_k)$, then stop and return $y^*=y_0$. Otherwise, let $m=m_0$ be a nonnegative integer and go to **Step 2**.

Step 2 : Set $z=z_0+2^m$ and $y=d\cdot z$. If $\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i\notin\Theta$ and $m=0$, then stop and return $y^*=y_0$; if $\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i\notin\Theta$ and $m>0$, then let $m=m-1$ and go back to **Step 2**; otherwise, evaluate $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)$.

Step 3 : If $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)<\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i,N_k)$ and $z<z^{\max}$, then let $z_0=z$ and go to **Step 2**. If $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)<\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i,N_k)$ and $z\geq z^{\max}$, then stop and return $y^*=y$. If $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)\geq\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i,N_k)$ and $m=0$, then stop and return $y^*=y_0$. If $\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y\mathbf{e}_i,N_k)\geq\bar{G}(\hat{\mathbf{x}}^*_{k-1}+y_0\mathbf{e}_i,N_k)$ and $m>0$, then let $m=m-1$ and go to **Step 2**.

In Algorithm 2 the descent direction is $d\mathbf{e}_i$. The quantity z is the distance (always nonnegative) along the descent direction from $\hat{\mathbf{x}}^*_{k-1}$, the quantity y is z if the descent direction is $\mathbf{e}_i$ and $-z$ if the descent direction is $-\mathbf{e}_i$. The constant m_0 controls 2^{m_0}, which is the maximum step size the line search may take. It should be set according to the size of the feasible region of the problem. The constant $z^{\max}$ controls the maximum distance from $\hat{\mathbf{x}}^*_{k-1}$ and $z\leq z^{\max}+2^{m_0}$. This ensures that Algorithm 2 satisfies Condition 2. We suggest to set $z^{\max}$ as some large number.

When Algorithm 2 is used for solve Problem (2) in Algorithm 1, Conditions 1–3 are all satisfied.

4 ASYMPTOTIC PROPERTIES OF COORDINATE SEARCH

In this section we give the asymptotic properties of Algorithm 1, including local convergence and limiting distributions. The proofs of the theorems are omitted in this paper, they follow from the theorems in Hong and Nelson (2005).

4.1 Local Convergence

When Problem (1) is fully constrained, Θ is a finite set. Therefore, $\hat{\mathbf{x}}^*_k$ equals to some solutions in Θ infinite often. We are able to show that, according to Hong and Nelson (2005), those solutions are all local minimizers to Problem (1). The convergence result for fully constrained problem is summarized in the following theorem:

Theorem 1 *If Assumption 1 and Condition 1 are satisfied, and if $N_k\to\infty$ as $k\to\infty$, then the infinite sequence $\{\hat{\mathbf{x}}^*_0,\hat{\mathbf{x}}^*_1,\ldots\}$ generated by Algorithm 1 converges with probability 1 to $\mathcal{M}$ in the sense that* $\mathrm{P}\{\hat{\mathbf{x}}^*_k\notin\mathcal{M}$ infinitely often$\}=0$.

When Problem (1) is partially constrained or unconstrained, Θ includes a countably infinite number of solutions. By Condition 2, only a finite number of solutions are visited in each iteration. Since $\mathbf{x}_0$ is included in the comparison in each iteration of Algorithm 1, then we can show that $\hat{\mathbf{x}}^*_k$ may be outside of the level set $\mathcal{L}$ at most a finite number of times. Since $\mathcal{L}$ is a finite set according to Assumption 2, the partially constrained or unconstrained problem behaves like a fully constrained problem asymptotically. Therefore, the conclusion of Theorem 1 also holds. The convergence result for partially constrained or unconstrained problem is summarized in the following theorem:

Theorem 2 *If Assumptions 1 and 2 and Conditions 1 and 2 are satisfied, and if $N_k\to\infty$ as $k\to\infty$, then the infinite sequence $\{\hat{\mathbf{x}}^*_0,\hat{\mathbf{x}}^*_1,\ldots\}$ generated by Algorithm 1 converges with probability 1 to $\mathcal{M}$ in the sense that* $\mathrm{P}\{\hat{\mathbf{x}}^*_k\notin\mathcal{M}$ infinitely often$\}=0$.

For the partially constrained or unconstrained problems, the coordinate search requires fewer assumptions compared to COMPASS algorithm of Hong and Nelson (2004). Since COMPASS cannot guarantee that the number of solutions visited in each iteration is bounded above, it requires that the number of simulation observations allocated to a solution visited in iteration k, which is similar to N_k in Algorithm 1, to grow above certain rate. In coordinate search, however, line search algorithms, e.g., Algorithm 2, may guarantee that the number of solutions visited in each iteration is bounded above (as required in Condition 2).

4.2 Limiting Distributions

To analyze the limiting distributions of the coordinate search, we make the following assumptions:

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

Assumption 3 is more restrictive than Assumption 1. If $G_i(\mathbf{x})$, $i=1,2,\ldots$, satisfy Assumption 3, they also satisfy Assumption 1 according to the Strong Law of Large Numbers. Assumption 3 is used to show that the sequence of simulation observations obtained at the local minimizer satisfies a central limit theorem.

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

If there are more than one local minimizer, Algorithm 1 may converge to any of them and it is difficult to characterize the probability of converging to a particular local minimizer. By making Assumption 4, we focus on the problem with only one local minimizer and we know the algorithm converges to this solution. Though Assumption 2 is restrictive, we believe that it also provides some insights on the problems with multiple local minimizers.

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

Theorem 3 *When Algorithm 1 is applied to solve Problem (1), if Assumptions 3 and 4 are satisfied when the problem is fully constrained, or Assumptions 2–4 are satisifed when the problem is partially constrained or unconstrained, and if there exists a deterministic sequence* $\{c_k\}$ *such that* $N_k/c_k \to 1$ *and* $c_k \to \infty$ *as* $k \to \infty$*, then*

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

where $Z \sim N(0,1)$.

When Problem (1) has only one local minimizer $\mathbf{x}^*$, by Theorems 1 and 2 we know that $\hat{\mathbf{x}}_k^* \neq \mathbf{x}^*$ for only a finite number of k with probability 1 for both fully constrained problem and partially constrained or unconstrained problem. Therefore, showing Theorem 3 is equivalent of showing that

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

where is a direct result of Assumption 3 and the standard Central Limit Theorem.

Theorem 3 gives the limiting distribution in terms of the simulation effort allocated to $\hat{\mathbf{x}}_k^*$. However, it is more interesting to know what the limiting distribution is in terms of the total simulation effort. Let n_k denote the total number of simulation observations allocated to all solutions through iteration k, n_k represents the total amount of computational effort that has spent to solve the problem through iteration k. The following corollary gives the conditions under which we are able to find the limiting distribution.

Corollary 1 *In addition to the conditions of Theorem 3, if there exists a constant c such that* $n_k/N_k \to c$ *in probability as* $k \to \infty$*, then*

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

When Condition 3 is satisfied by the line search algorithm, e.g., Algorithm 2, only the solutions in $\mathcal{N}(\mathbf{x}^*)$ are visited infinite often besides $\mathbf{x}^*$ for fully constrained problems. For partially constrained or unconstrained problems, besides solutions in $\mathcal{N}(\mathbf{x}^*)$ and $\mathbf{x}^*$, $\mathbf{x}_0$ is also visited infinitely often. If $N_k/N_{k-1} \to 1$ as $k \to \infty$, which means that N_k does not increase by orders of magnitute, then we are able to show that, for fully constrained problems,

$$n_k/N_k \to |\mathcal{N}(\mathbf{x}^*)| + 1 \quad \text{in probability}$$

and, for partially constrained or unconstrained problems,

$$n_k/N_k \to |\mathcal{N}(\mathbf{x}^*)| + 2 \quad \text{in probability.}$$

Therefore, the variance of the limiting distribution is either $(|\mathcal{N}(\mathbf{x}^*)| + 1)\sigma^2(\mathbf{x}^*)$ or $(|\mathcal{N}(\mathbf{x}^*)| + 2)\sigma^2(\mathbf{x}^*)$.

When globally convergent algorithms are applied to solve Problem (1) and if the problem is fully constrained, then the globally convergent algorithms guarantee the global convergence by showing that all solutions are visited infinitely often. Andradóttir (1999) provides the limiting distribution for the algorithms. If in the limit that all solutions obtain the same number of simulation observations and if the problem has only one local minimizer, then

$$\sqrt{n_k}\left[\bar{G}\left(\hat{\mathbf{x}}_k^*, N_k\right) - g(\mathbf{x}^*)\right] \Rightarrow \sqrt{|\Theta|}\ \sigma(\mathbf{x}^*) \cdot Z \quad \text{as } k \to \infty.$$

The variance of the limiting distribution is $|\Theta|\sigma^2(\mathbf{x}^*)$ which may be much larger than $(|\mathcal{N}(\mathbf{x}^*)| + 1)\sigma^2(\mathbf{x}^*)$ when the size of problem is large. This explains the advantage of local convergence and locally convergent algorithms. For partially constrained or unconstrained problems, no limiting distributions have been provided for the globally convergent algorithms.

5 NUMERICAL RESULTS

In this section we compare the performance of the coordinate search to the performance of COMPASS. We use Algorithm 2 to solve the one-dimensional subproblems in the coordinate search. We have conduct a number of simulation experiments, and we first summarize our findings as follows:

- COMPASS is a random search algorithm. It generally performs well at the beginning of the search since it enables the algorithm to jump far away from the starting point since the probability of finding a better solution than the starting solution is often large at the beginning of the search. When the sample best solution is getting closer to the optimal solution, however, the probability of finding better solutions may get lower. Then COMPASS may not find a better solution for a large number of consecutive iterations. Coordinate search is a deterministic algorithm and it looks for better solutions systematically. It may not be as good as COMPASS at the beginning. However, it keeps improving the objective values almost in all iterations until finding optimal solutions.

- COMPASS simulates all visited solutions in each iteration. As the number of visited solutions become large, the simulation observations required in each iteration becomes larger. This significantly slows down the search process of COMPASS. Coordinate search, however, does not simulate the irrelevant solutions in each iteration. It spends much fewer simulation observations in each iteration when iteration count becomes large compared to COMPASS. This also explains why coordinate search has better performance later in the search.
- In each iteration, COMPASS constructs a most promising area using all visited solutions. Then it uses a Markov-chain based algorithm to sample uniformly from the solutions in the area. The sampling algorithm requires a significant computational overhead especially when the number of visited solutions becomes large. Compared to COMPASS, coordinate search requires less computational overhead.
- COMPASS is constructed based on the geometric properties of the Euclidean space. It generally works well for small dimensional problems since the geometry of low dimensional space can be clearly pictured. When the dimension of the problem becomes large, however, the geometry becomes more intriguing and less intuitive. It seems that COMPASS is not good at solving problems over 20 dimensions. Coordinate search, however, searches along each coordinate direction. It works well even for DOvS problems of more than 20 dimensions.

In the rest of the section we compare the performances of coordinate search and COMPASS through two examples: a traditional (s, S) inventory problem and a thirty-dimensional quadratic function.

5.1 (s, S) Inventory Problem

Consider a classic (s, S) inventory problem (Koenig and Law 1985 and Hong and Nelson 2004) 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 $S - s \geq 10$, $20 \leq s \leq 80$, $40 \leq S \leq 100$, and $s, S \in \mathcal{Z}$. The optimal inventory policy is $(20, 53)$ with expected cost per 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.

Figure 1 shows the performances of COMPASS and coordinate search when applied to solve the problem. Each curve in the figure represents the average of 50 sample paths

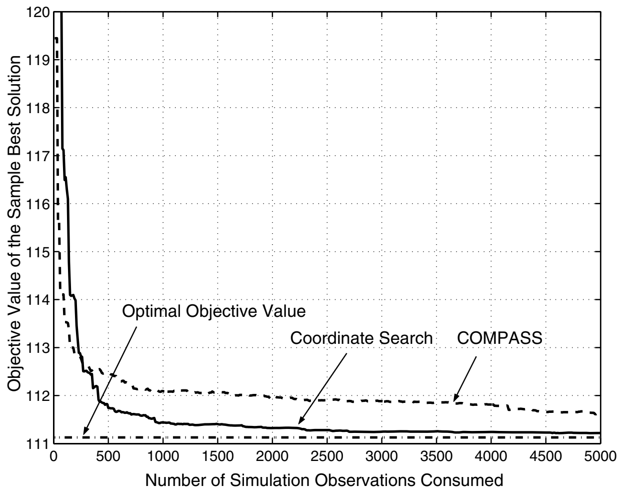

Figure 1: Average Performances of COMPASS and Coordinate Search when Solving (s, S) Inventory Problem

(realization of the algorithm). From the figure we can see that COMPASS performs well at the beginning and coordinate search soon catches up and finds optimal solution or solutions that close to the optimal solution more quickly. Since the differences between the objective values of the optimal solution and its neighboring solutions are very small, the noise in the function evaluations dominates the differences. Therefore, coordinate search does not always find the optimal solution within 5000 simulation observations.

5.2 A Thirty-Dimensional Quadratic Problem

Consider a 30-dimensional quadratic function, $g(\mathbf{x}) = x_1^2 + x_2^2 + \cdots + x_{30}^2 + 1$. Let $G(\mathbf{x}) = g(\mathbf{x}) + \epsilon(\mathbf{x})$ and $\epsilon(\mathbf{x})$ has a normal distribution with mean 0 and standard deviation $0.05g(\mathbf{x})$, and we let $-100 \leq x_i \leq 100$, $i = 1, 2, \ldots, 30$. The optimal solution of the problem is $x_i = 0$, $i = 1, 2, \ldots, 30$, with the optimal objective value 1. Notice that the problem has about 1.25×10^{69} feasible solutions. We set the starting point as $x_i = 80$, $i = 1, 2, \ldots, 30$. The objective value of the starting point is 192000 with the standard deviation 9600. The noise dominates the differences of the objective values of the starting point and its neighboring solutions.

Figure 2 shows the performances of COMPASS and coordinate search when applied to solve the problem. Each curve in the figure represents the average of 50 sample paths. From the figure we see that the coordinate search performs better. Among all 50 sample paths, 49 of them find the optimal solution and one has an objective value 9. From Figure 3 we can see that the average objective value is below 10 after consuming 60000 simulation observations. We also tried DOvS problems with even larger dimensions, coordinate search may also solve them by a reasonable number of simulation observations.

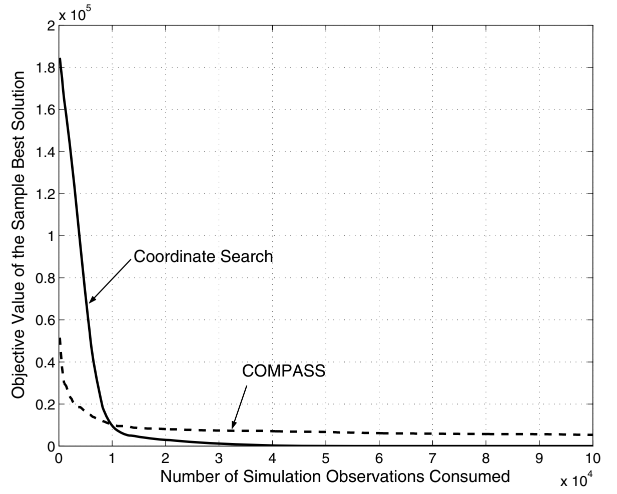

Figure 2: Average Performances of COMPASS and Coordinate Search when Solving the 30-dimensional Quadratic Function

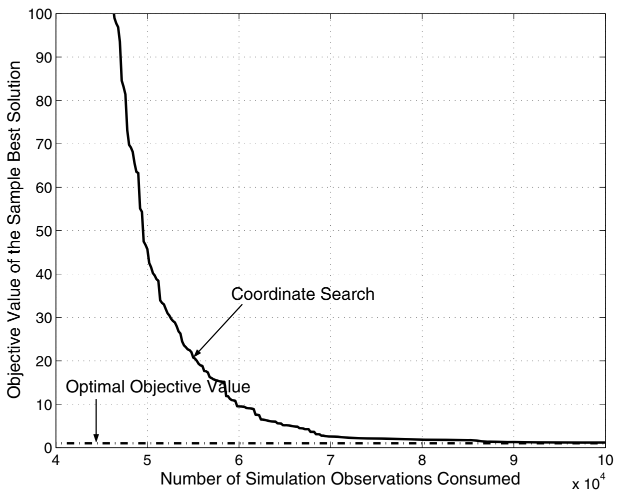

Figure 3: Average Performance of COMPASS when Solving the 30-dimensional Quadratic Function

ACKNOWLEDGMENT

This research was partially supported by the Hong Kong Research Grants Council grant number DAG04/05.EG04.

REFERENCES

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

Andradóttir, S. 1999. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, 9:349–380.

April, J., M. Better, F. Glover and J. Kelly. 2004. New advances and applications for marrying simulation and optimization. *Proceedings of the 2004 Winter Simulation Conference*, 80–86.

Baras, J. S. 2003. Modeling and simulation of telecommunication networks for control and management. *Proceedings of the 2003 Winter Simulation Conference*, 431–440.

Fu, M. C. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14:192–215.

Gerencsér, L., S. D. Hill and Z. Vágó. 1999. Optimization over discrete sets via SPSA. *Proceedings of the 38th Conference on Decision and Control*, 1791–1795.

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

Hong, L. J. and B. L. Nelson. 2004. Discrete Optimization via Simulation using COMPASS. *Operations Research*, forthcoming.

Hong, L. J. and B. L. Nelson. 2005. A framework of locally convergent algorithms for solving discrete optimization-via-simulation problems. Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Working Paper.

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

Koenig, L. W., and 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. *Communications in Statistics: Simulation and Computation*,14:719–734.

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

Truong, T. H. and F. Azadivar. 2003. Simulation based optimization for supply chain configuration design. *Proceedings of the 2003 Winter Simulation Conference*, 1268–1275.

Vogt, H. 2004. A New Method to Determine the Tool Count of a Semiconductor Factory Using FabSim. *Proceedings of the 2004 Winter Simulation Conference*, 1925–1929.

Wieland, F. and T. C. Holden. 2003. Targeting aviation delay through simulation optimization. *Proceedings of the 2003 Winter Simulation Conference*, 578–584.

Yan, D. and H. Mukai. 1992. Stochastic discrete optimization. *SIAM Journal of Control and Optimization*, 30:594–612.

AUTHOR BIOGRAPHY

L. JEFF HONG is an assistant professor in the Department of Industrial Engineering and Logistics Management at The Hong Kong University of Science and Technology. He received his Ph.D. in Industrial Engineering and Management Sciences from Northwestern University in 2004 and his research interests include optimization via simulation and simulation ranking and selection. His e-mail address is <hongl@ust.hk> and his web page is <ihome.ust.hk/~hongl>.