*Proceedings of the 2002 Winter Simulation Conference*
*E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.*

# A COMBINED PROCEDURE FOR OPTIMIZATION VIA SIMULATION

Juta Pichitlamken

Department of Industrial Engineering
Kasetsart University
Bangkok, THAILAND

Barry L. Nelson

Department of Industrial Engineering
& Management Sciences
Northwestern University
Evanston, IL 60208-3119, U.S.A.

## ABSTRACT

We propose an optimization-via-simulation algorithm for use when the performance measure is estimated via a stochastic, discrete-event simulation, and the decision variables may be subject to deterministic linear integer constraints. Our approach—which consists of a global guidance system, a selection-of-the-best procedure, and local improvement—is globally convergent under very mild conditions.

## 1 INTRODUCTION

We consider the following optimization problem:

$$\max_{\mathbf{x}\in\Theta} \mu(\mathbf{x})$$

where $\mu(\mathbf{x})$ is the scalar-valued performance measure associated with a vector-valued, integer decision variable or *solution* $\mathbf{x} = (x_1, x_2, \ldots, x_q)$. All solutions $\mathbf{x}$ are contained in a convex and finite feasible space $\Theta$, defined by deterministic linear constraints. We assume the performance measure $\mu(\mathbf{x})$ can only be estimated via a stochastic, discrete-event simulation by observing a simulation output random variable $Y(\mathbf{x})$, where $\mu(\mathbf{x}) = \mathrm{E}[Y(\mathbf{x})]$, and that little or nothing is known about the response surface $\mu(\mathbf{x})$.

A key feature that makes optimization via simulation difficult is balancing the tradeoff between the computational effort used in estimating $\mu(\mathbf{x})$ and that used for exploration of $\Theta$ in search of better solutions. If too much effort is spent on estimating $\mu(\mathbf{x})$, an optimization algorithm may not visit much of $\Theta$ in the time available. As a result, it may not discover an optimal solution, or even a good solution, at all. On the other hand, unless the algorithm accounts for variability in the estimator of $\mu(\mathbf{x})$, the search can be misled and not converge or even recognize good solutions when they are encountered. Our goal is to achieve both *provable convergence* and *good empirical performance*.

Our search scheme consists of a global guidance system, a selection-of-the-best procedure, and local improvement. The global guidance system ensures the convergence of the search so that, given sufficient time, it reaches and selects one of the optimal solutions. Specifically, we adopt the philosophy of Shi and Ólafsson's (2000) *Nested Partition* (NP) method. NP is based on identifying a sequence of "most-promising" subregions of $\Theta$. When better solutions are found *inside* the current most-promising region, then the region is partitioned for finer exploration. On the other hand, when better solutions are found *outside* the current most-promising subregion, then NP backtracks to a superregion of it. The idea is to concentrate the computational effort where there appear to be good solutions but not be trapped locally.

The search, essentially the partitioning, is guided by the estimated performance of solutions sampled from each region. Thus, the performance of an NP-based search depends, to a great extent, on making correct decisions about when to partition or backtrack. We incorporate *Sequential Selection with Memory* (SSM)—a statistical procedure that we have specifically designed for use within the optimization-via-simulation context—to help NP make such decisions (Pichitlamken and Nelson 2001). Under certain conditions, SSM guarantees to select the best, or a near-best, solution with a user-specified probability when some solutions have previously been visited and their past observations (or their summary statistics) are maintained. SSM utilizes "memory" of solutions it has seen to alleviate the need to obtain new simulation outputs every time the search revisits a solution. We further refine SSM for use in an NP-based search by allowing it to terminate as soon as the most-promising region, rather than the single best solution, is identified. We call this refinement SSM(`REGION`).

Our motivation for incorporating SSM is to intelligently expend the simulation effort used in region-selection steps. The hope is that good selections made with minimal simulation effort, and the guidance provided by the NP method, will result in moving toward better and better solutions, while still having adequate time to explore $\Theta$ thoroughly.

We further enhance the performance of our combined scheme with local improvement. Because the NP method already provides a *diversification* element, local improvement is intended to provide an *intensification* component. The idea is to enhance performance on problems where $|\Theta|$ is large, but good solutions are clustered, or where $|\Theta|$ is large, but the response surface is smooth. When $|\Theta|$ is large, relative to the time available for optimization, the search may miss some better solutions residing close to good solutions that it has already visited. Once they are missed, the search may not encounter them again due to random sampling in a large space. Local improvement helps the search explore $\Theta$ more intensively near good solutions.

A hill-climbing (HC) algorithm constitutes our local-improvement scheme. We chose HC because it is intuitively simple: The current solution on hand is compared with some (or all) of its neighboring solutions, and the winner becomes the next solution. This neighborhood selection of the best is repeated until some stopping criterion is satisfied. Its simplicity aside, HC is also appealing because it is readily applicable in our problem setting where a neighborhood is easy to define.

With the NP method acting as our global guidance system, SSM expediting the region selection, and HC performing local improvement, we call our combined scheme the *NP+SSM+HC Algorithm*. In addition to the use of SSM and HC, our implementation of NP differs considerably from Shi and Ólafsson's (2000) version, including the criterion used to estimate the optimal solution, the generality of the partitioning scheme and the solution-sampling scheme. Similar to their NP method, NP+SSM+HC converges almost surely to a global optimum, but it does so under far less restrictive assumptions.

The paper is organized as follows: The next section reviews literature relevant to our approach. We define our problem more specifically in Section 3. We give an overview of NP+SSM+HC in Section 4, followed by descriptions of each of its components. The global convergence properties are stated in Section 5. We compare our algorithms to other schemes via a numerical example in Section 6. We conclude with future research directions in Section 7. All the proofs can be found in Pichitlamken (2002).

## 2 BACKGROUND

Many optimization-via-simulation algorithms are adapted from methods designed for deterministic problems. Typically, the search tries to move in a relatively improving direction while utilizing some form of randomization to escape from local optimal solutions. Andradóttir's (1998) tutorial discusses a number of such methods, including the stochastic ruler algorithm (Yan and Mukai 1992), variants of simulated annealing altered to accommodate randomness (e.g., Gelfand and Mitter 1989, Gutjahr and Pflug 1996, and Alrefaei and Andradóttir 1999), and Andradóttir's (1995, 1996) random search algorithms. The common characteristic among these algorithms is that they move from the current solution to one of its neighboring solutions. However, they differ in their neighborhood structure, their rule for comparing the current solution to the selected neighbor, and their criterion for estimating the optimal solution at the end of the search. Most of these algorithms can be shown to converge globally as the sampling effort increases.

In contrast to such globally convergent methods, many heuristics are appealing because they work well in practice. To bridge the gap between research and practice, Boesel (1999) and Boesel, Nelson and Ishii (2002) supplement a genetic algorithm search with a ranking-and-selection procedure applied at the end of the search to allow the combined algorithm to make a correct-selection guarantee.

## 3 FRAMEWORK

We now define our optimization-via-simulation problem more precisely. Our goal is to solve

$$\max_{\mathbf{x}\in\Theta} \mu(\mathbf{x}) \qquad (1)$$

when $\Theta$ is defined by the following constraints:

$$\begin{array}{rcrcll} & & \sum_{i=1}^{q} a_{ij}x_i & \leq & b_j, & j=1,2,\ldots,p \\ 0 \leq l_i & \leq & x_i & \leq & u_i < \infty, & i=1,2,\ldots,q \\ & & l_i, x_i, u_i & \in & \mathcal{Z}^+ \cup \{0\}, & i=1,2,\ldots,q \end{array} \qquad (2)$$

where the set of positive integers is denoted by $\mathcal{Z}^+$. Thus, we assume that the feasible region $\Theta$ is convex and finite. To avoid triviality, we also assume that $\Theta$ is nonempty. The finiteness of $\Theta$ allows us to index the solutions $\mathbf{x}$ and the corresponding performance measures as follows: $\Theta = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_v\}$ where $v$ is the number of feasible solutions in $\Theta$, and $\mu_i \equiv \mu(\mathbf{x}_i)$. Without loss of generality, we let the set of optimal solutions be $\Theta^* = \{\mathbf{x}_{v^*}, \mathbf{x}_{v^*+1}, \ldots, \mathbf{x}_v\}$ where $1 \leq v^* \leq v$.

Recall that we must estimate $\mu_i$. The observed performance measure of solution $i$ on replication $p$ of the simulation is denoted by $Y_{ip}$, so that $\mu_i = \mathrm{E}[Y_{ip}]$. Let $\sigma_i^2 = \mathrm{Var}\,[Y_{ip}]$. Both the performance measure $\mu_i$ and its variance $\sigma_i^2$ are assumed finite for all $i \in \{1, 2, \ldots, v\}$. Further, the observed performance measures $Y_{ip}$, $p = 1, 2, \ldots$, are independent and identially distributed (i.i.d.), and independent of $Y_{js}$ for $i \neq j$.

Let a *region*, which we will generically denote as $\tau$, be a finite, convex subset of $\Theta$ characterized by:

$$\begin{array}{rcll} \sum_{i=1}^{q} a_{ij}^{\tau} x_i & \leq & b_j^{\tau}, & j = 1, 2, \ldots, p^{\tau} \\ 0 \leq l_i^{\tau} \leq x_i & \leq & u_i^{\tau} < \infty, & i = 1, 2, \ldots, q \\ l_i^{\tau}, x_i, u_i^{\tau} & \in & \mathcal{Z}^+ \cup \{0\}, & i = 1, 2, \ldots, q. \end{array} \quad (3)$$

The region is *infeasible* when there exists a constraint $j \in \{1, 2, \ldots, p^{\tau}\}$ such that

$$\sum_{i: a_{ij}^{\tau} > 0} a_{ij}^{\tau} l_i^{\tau} + \sum_{i: a_{ij}^{\tau} < 0} a_{ij}^{\tau} u_i^{\tau} \quad > \quad b_j^{\tau}. \qquad (4)$$

## 4 NP+SSM+HC

We first give a high-level description of NP+SSM+HC, and then present the particulars of each step—partitioning, solution sampling, SSM, and HC—separately in the subsequent sections.

### Algorithm NP+SSM+HC

1. *Initialization:* Set the iteration counter $k = 1$, the current most-promising region $R_k = \Theta$, the number of observations on the $i^{\text{th}}$ solution $n_i(k) = 0$ for all $i \in \{1, 2, \ldots, v\}$, and the initial estimate of the optimal solution $\mathbf{x}_{\hat{i}^*_{k-1}}$ to a user-provided initial solution.
2. *Search and selection:* Repeat Steps 2a–2f until the simulation effort (i.e., clock time or the number of simulation replications allowed) is exhausted:
   (a) *Partitioning:* If the current most-promising region $R_k$ is not a singleton, then partition $R_k$ into disjoint regions $R_{k1}, R_{k2}, \ldots, R_{k\omega(R_k)}$ (see Section 4.1). Let $M_k = \omega$ be the number of subregions. Then, if $R_k \neq \Theta$, aggregate the surrounding region; let $M_k = M_k + 1$ and $R_{kM_k} = \Theta \setminus R_k$.
   (b) *Sampling:* For each region $R_{k\ell}, \ell = 1, 2, \ldots, M_k$, randomly sample $\vartheta$ solutions from $R_{k\ell}$. (If $\mathbf{x}_{\hat{i}^*_{k-1}} \in R_{k\ell}$, include it as one of these $\vartheta$ sampled solutions from $R_{k\ell}$. See Section 4.2.) Aggregate *all* the sampled solutions $\mathbf{x}_i$ into a set through their indices $i$; let $\mathcal{S}_k$ denote the set of indices of sampled solutions.
   (c) *Selection of the best solution:* Take $\Delta n_{\text{free}}$ observations of $Y_{ip}$ from every solution $\mathbf{x}_i, i \in \mathcal{S}_k$. Use SSM or SSM(`REGION`) to select the best solution over $\mathcal{S}_k$, which we denote as $\hat{\mathbf{x}}^*(\mathcal{S}_k)$ (see Section 4.3). If the simulation effort is exhausted, go to *Search termination* step.
   (d) *Algorithm Hill Climbing*: If the criterion for using HC is satisfied, perform Algorithm Hill Climbing with $\hat{\mathbf{x}}^*(\mathcal{S}_k)$ as a starting solution (see Section 4.4). Let $\mathbf{x}_{\hat{i}^*_k}$ be the solution deemed best by HC. If the simulation effort is exhausted, go to *Search termination* step.
   (e) *Updating the most-promising region:* If $\mathbf{x}_{\hat{i}^*_k} \in R_k$, then $R_{k\ell}$ that contains $\mathbf{x}_{\hat{i}^*_k}$ becomes the next most-promising region, $R_{k+1}$; otherwise, the search backtracks to the superregion of $R_k$, which can be either $\Theta$ or $R_{k-1}$. Increment $k = k + 1$.
   (f) *Restart:* Restart at iteration $k$ if $R_{k-k_0+1} = R_{k-k_0+2} = \cdots = R_k$ by letting $R_k = \Theta$ and using a different partitioning criterion (see (7) below).
3. *Search termination:* The best solution selected by NP+SSM+HC is the one with the maximum cumulative sample average; i.e., the selected solution is $\mathbf{x}_{\hat{i}^*}$ where

$$\hat{i}^* \equiv \arg\max_{1 \leq i \leq v} \left\{ \bar{Y}_i(n_i(k)) : n_i(k) > 0 \right\} (5)$$

$$\bar{Y}_i(r) \equiv \sum_{p=1}^{r} Y_{ip}/r. \qquad (6)$$

We further describe each component of NP+SSM+HC in the following sections (see Pichitlamken 2002 for complete details).

### 4.1 Partitioning Scheme

Our proposed partitioning scheme is motivated by the Branch and Bound Method. The goal is to partition a convex, feasible region $\tau$ of the form defined in (3) into disjoint subregions, each of which remains convex. We branch $\tau$ in one dimension per partitioning: First, we select the variable to branch upon, say $x_{i'}$, using one of the following three criteria:

$$i' \equiv \begin{cases} \text{argmax}_{1 \leq i \leq q} \{u_i^{\tau} - l_i^{\tau}\} \text{ if criterion = } \texttt{BIGGEST RANGE} \\ \text{argmin}_{1 \leq i \leq q} \{u_i^{\tau} - l_i^{\tau}\} \text{ if criterion = } \texttt{SMALLEST RANGE} \\ i \text{ with probability } 1/q \text{ for } i \in \{1, 2, \ldots, q\} \\ \quad \text{if criterion = } \texttt{RANDOM}. \end{cases} \qquad (7)$$

Next, we further divide the range of feasible values of $x_{i'}$, $\{l_{i'}^{\tau}, l_{i'}^{\tau} + 1, \ldots, u_{i'}^{\tau} - 1, u_{i'}^{\tau}\}$, into subintervals. Based on these subintervals, we form the subregions from the constraints that define $\tau$ and a tighter constraint on $x_{i'}$.

On iteration $k$, after the most-promising region $R_k$ is partitioned, and the surrounding region is aggregated, solutions are sampled from each subregion. We describe the solution sampling scheme next.

### 4.2 Solution Sampling Scheme

NP+SSM+HC requires sampling within the subsets of the most-promising region $R_k$, which are convex, and within the surrounding region, which is not convex. We developed Algorithm MIX-D to sample an integer solution from a convex region of the form (3) (we will denote it as $\tau$ for simplicity), and Algorithm MIX-DS to sample an integer solution from $\Theta \setminus R_k$. Algorithm MIX-DS is essentially Algorithm MIX-D, sampling over $\Theta$, but with extra calculations to ensure that the generated Markov chain remains outside $R_k$ at all times.

Our discrete-variable sampling algorithms extend Smith's (1984) Mixing Algorithm, which is for continuous spaces. From a starting solution inside $\tau$, Algorithm MIX-D generates the next solution (i.e., the next state of the Markov chain) that is also inside $\tau$. This process is repeated for several steps, and the sampled solution returned by MIX-D is the current state of the Markov chain when it stops. If the feasible regions $\Theta$ and $\tau$ are of the form (3), and they are nonempty and of full-dimension, both MIX-D and MIX-DS generate solutions that are asymptotically uniformly distributed over the feasible space $\tau$ and $\Theta \setminus R_k$, respectively (see Pichitlamken 2002 for details).

Although the uniformity property of the solution sampling algorithms is not required to attain the global convergence of NP+SSM+HC, it is desirable in practice: Recall that we assume no knowledge of the response surface $\mu(\mathbf{x})$. Therefore, we count on the MIX-D and MIX-DS algorithms to insure diversity, while the NP method focuses NP+SSM+HC on promising solutions.

After we sample solutions from all subregions, we use either SSM or SSM(`REGION`) to select the best. To make this paper self-contained, we briefly describe these procedures in the next section (see Pichitlamken and Nelson 2001 and Pichitlamken 2002 for complete details).

### 4.3 Sequential Selection with Memory

SSM was designed to provide a highly efficient method for selecting the best—maximum of minimum expected performance—from among a small number of candidate solutions. SSM is fully sequential with elimination, which means that it takes observations one at a time from the solutions under consideration and eliminates (ceases sampling) solutions as soon they are shown to be inferior. SSM is specially designed for use in optimization algorithms that revisit solutions because it exploits whatever data have already been obtained. In NP+SSM+HC, we use SSM for the selection of a new most-promising region for NP, and in determining when the HC algorithm has found an improved solution.

Without loss of generality, let the finite number of solutions under consideration be denoted by $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\kappa\}$. SSM assumes the observations taken from $\mathbf{x}_i$, $Y_{ip}$, to be i.i.d. normally distributed with finite mean $\mu_i$ and variance $\sigma_i^2$ (the convergence properties of NP+SSM+HC do not depend on the normality assumption). For convenience of exposition, assume that the true means of the solutions are indexed such that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_\kappa$. The best solution is defined as the one with the largest mean, which is therefore $\mu_\kappa$.

Our procedure guarantees to select $\mathbf{x}_\kappa$ with probability at least $1-\alpha$ whenever the difference between the best and the next best solution is worth detecting:

$$\Pr\{\text{select } \mathbf{x}_\kappa\} \geq 1-\alpha \text{ whenever } \mu_\kappa - \mu_{\kappa-1} \geq \delta. \quad (8)$$

The indifference-zone parameter is denoted by $\delta > 0$. Even when the indifference-zone condition is not satisfied ($\mu_\kappa - \mu_{\kappa-1} < \delta$), SSM still selects a "good" solution (one whose mean is within $\delta$ of $\mu_\kappa$) with probability at least $1-\alpha$.

In SSM, we sequentially take one observation at a time from *surviving* solutions, immediately followed by screening. To make this precise, we define some notation: Let $i \in \{1, 2, \ldots, \kappa\}$.

$$\begin{aligned}
V &= \text{set of solutions we have "visited" before;} \\
&\phantom{=} n_i \geq n_0 \text{ for } i \in V \\
V^c &= \text{set of solutions we see for the first time;} \\
&\phantom{=} n_i < n_0 \text{ for } i \in V^c = \{1,2,\ldots,\kappa\} \setminus V \\
N_{ij} &= \left\lfloor \frac{a_{ij}}{\lambda} \right\rfloor \\
N_i &= \max_{j \neq i}\{N_{ij}\} \\
N &= \max_{1 \leq i \leq \kappa} N_i \qquad (9) \\
\sigma_{ij}^2 &= \text{Var } (Y_{ip} - Y_{jp}) \\
n_0 &= \text{minimum initial number of observations} \\
&\phantom{=} \text{from any solution} \\
S_{ij}^2 &= \text{estimator of } \sigma_{ij}^2 \\
&= \frac{1}{n_0 - 1} \sum_{p=1}^{n_0} \left(Y_{ip} - Y_{jp} - [\bar{Y}_i(n_0) - \bar{Y}_j(n_0)]\right)^2 \\
f &= n_0 - 1.
\end{aligned}$$

Note that $N_i + 1$ is the maximum number of observations taken from $\mathbf{x}_i$.

**Procedure SSM**

1. *Initialization:* For each $\mathbf{x}_i, i \in V^c$, take $n_0 - n_i$ observations ($n_0 \geq 2$), and set $n_i = n_0$. Compute $S_{ij}^2, \forall i \neq j$.
2. *Procedure parameters:* We choose $\lambda$ and $a_{ij}$ as follows:

$$\lambda = \frac{\delta}{2} \quad \text{and} \quad a_{ij} = \frac{f S_{ij}^2}{4(\delta - \lambda)} \left[ \left( \frac{\kappa - 1}{2\alpha} \right)^{2/f} - 1 \right].$$

   If $n_0 > N$ (as defined in (9)), stop and select the solution with the largest $\bar{Y}_i(n_i)$ (as defined in (6)) as the best. Otherwise, let $I = \{1, 2, \ldots, \kappa\}$ be the set of surviving solutions, set $r = n_0$, and proceed to *Screening*. From here on $V$ represents the set of solutions on which we have obtained more than $r$ observations, while $V^c$ is the set of solutions with exactly $r$ observations.
3. *Screening:* Set $I^{\text{old}} = I$ where

$$\begin{aligned} I &= \Big\{ i : i \in I^{\text{old}} \text{ and} \\ &\qquad \mathcal{Y}_i \geq \max_{j \in I^{\text{old}}, j \neq i} \left( \mathcal{Y}_j - a_{ij} \right) + r\lambda \Big\} \\ \mathcal{Y}_j &= \begin{cases} \sum_{p=1}^{r} Y_{jp} & \text{for } j \in V^c \\ r\bar{Y}_j(n_j) & \text{for } j \in V. \end{cases} \end{aligned}$$

   In essence, for $\mathbf{x}_i$ with $n_i > r$, we substitute $r\bar{Y}_i(n_i)$ for $\sum_{p=1}^{r} Y_{ip}$.
4. *Stopping rule:* If $|I| = 1$, then stop and report the only survivor as the best; otherwise, for each $i \in (I \cap V^c)$, take one additional observation from $\mathbf{x}_i$ and set $r = r + 1$. If $r = N + 1$, terminate the procedure and select the solution in $I$ with the largest sample average as the best; otherwise, for each $i \in (I \cap V)$ with $n_i = r$, update $V$ and $V^c$: $V^c = V^c \cup \{i\}$, and $V = V \setminus \{i\}$. Go to *Screening*.

SSM(`REGION`) intends to save simulation effort by terminating SSM when all surviving solutions belong to the *same* subregion. This is useful in an NP step where all we need to do is to identify the subregion that contains the best sampled solution, not necessarily the best solution itself.

Let $\mathcal{S}_k^*$ be the set of solution indices of the surviving solutions when SSM(`REGION`) terminates on iteration $k$. The best solution selected by SSM(`REGION`) is:

$$\hat{\mathbf{x}}^*(\mathcal{S}_k) = \left\{ \mathbf{x}_i : i = \text{argmax}_{i \in \mathcal{S}_k^*} \bar{Y}_i(n_i) \right\}. \tag{10}$$

After SSM selects the best solution from the set of sampled solution, we attempt to improve it further with hill climbing, as described below.

### 4.4 Local Improvement

Algorithm Hill Climbing (HC) is essentially a greedy heuristic that iteratively moves from the current solution to one of its neighboring solutions until some stopping criterion is satisfied. In our algorithm, the available options for employing HC are: (a) do not perform HC at all, (b) perform HC on *all* iterations, or (c) perform HC if the *improvement* is “big enough," i.e.,

$$\left| \bar{Y}_{\hat{i}_k^*}(n_{\hat{i}_k^*}) - \bar{Y}_{\hat{i}_{k-1}^*}(n_{\hat{i}_{k-1}^*}) \right| \quad > \quad 2\delta, \tag{11}$$

where $\delta$ is an indifference-zone parameter of SSM (see (8)).

**Procedure Hill Climbing**

1. *Initialization:* Set $t = 0$ and $\mathbf{X}_0$ to a starting solution.
2. *Search:* Repeat Steps 2a–2d until the stopping criterion is satisfied (see Remark below):
   (a) *Neighborhood construction:* For $\mathbf{X}_t = (X_{1t}, X_{2t}, \ldots, X_{qt})$, construct a hypercube: $\mathcal{B}(\mathbf{X}_t) \equiv \Pi_{i=1}^{q}[X_{it} - \xi, X_{it} + \xi]$, where $\xi \in \mathcal{Z}^+$. Update $\mathcal{B}(\mathbf{X}_t)$ so that $\mathcal{B}(\mathbf{X}_t) \subseteq \Theta$.
   (b) *Solution sampling:* Use MIX-D (see Section 4.2) to sample $\varpi$ solutions from $\mathcal{B}(\mathbf{X}_t)$. Aggregate the indices of the sampled solutions into a set $\mathcal{S}_t^{\text{loc}}$.
   (c) *Selection of the best:* Use SSM (see Section 4.3) to select the best solution whose index is $\hat{i}_t^{*\text{loc}}$ from $\mathcal{S}_t^{\text{loc}}$.
   (d) *Update the best solution:* $\mathbf{X}_{t+1} = \mathbf{x}_{\hat{i}_t^{*\text{loc}}}$ and $t = t + 1$.
3. *Termination:* Return $\mathbf{X}_t$.

**Remark:** The options for the stopping criterion are: (a) to perform HC once, (b) to perform HC until $\hat{i}_t^{*\text{loc}} = \hat{i}_{t-1}^{*\text{loc}}$, or (c) to perform HC until

$$\left| \bar{Y}_{\hat{i}_t^{*\text{loc}}}(n_{\hat{i}_t^{*\text{loc}}}) - \bar{Y}_{\hat{i}_{t-1}^{*\text{loc}}}(n_{\hat{i}_{t-1}^{*\text{loc}}}) \right| \quad < \quad \delta, \tag{12}$$

where $\delta$ is an indifference-zone parameter. The motivation for (12) is similar to the rationale behind (11); with (12), however, HC stops if the perceived progress is too small.

Thus far, we have fully described how NP+SSM+HC works. In the next section, we establish its global convergence properties, which are not just an aggregation of the attributes of each component—partitioning, solution sampling, SSM, HC, and updating of the most-promising region—but also a result of their interactions.

## 5 PROPERTIES OF NP+SSM+HC

In Lemma 1, we establish that every solution in $\Theta$ is sampled infinitely often in the limit. The Strong Law of Large Numbers then leads us to almost-sure convergence of NP+SSM+HC.

**Lemma 1** *For all $i \in \{1, 2, \ldots, v\}$,*

$$\lim_{k\to\infty} n_i(k) = \infty.$$

□

**Theorem 1** *NP+SSM+HC converges almost surely to one of the global optimal solutions as $k \to \infty$; i.e., a solution $\mathbf{x}_{\hat{i}^*}$, where $\hat{i}^*$ as defined in (5) belongs to the set of optimal solutions almost surely as $k \to \infty$.* □

Theorem 1 guarantees that our algorithm converges to a global optimal solution in the limit. In Theorem 2, Andradóttir (1999) provides us with an $O(k^{-1/2})$ convergence rate for the case of a unique optimum ($v^* = v$). Note that Theorem 1 holds when there are multiple optimal solutions, but in such cases we were unable to extend Andradóttir's results to our algorithm. Theorem 2 also allows us to form a confidence interval for the maximum performance measure $\mu_v$.

**Theorem 2** *If the optimal solution is unique, and $\hat{i}^*$ satisfies (5), then*

$$\sqrt{\Delta n_{free} k}\left(\bar{Y}_{\hat{i}^*}\left(n_{\hat{i}^*}(k)\right) - \mu_v\right) \quad \Rightarrow \quad N(0, \sigma_v^2),$$

*where $\Rightarrow$ denotes convergence in distribution.* □

## 6 NUMERICAL EXPERIMENTS

We consider the performance of NP+SSM+HC relative to other optimization schemes on a three-stage buffer allocation problem. We will first describe the competing optimization schemes, then characterize a test problem, and finally report the corresponding performance of each scheme. In addition to NP+SSM+HC, the optimization approaches that we consider are:

**NP:** NP does not use SSM or HC. It takes $\Delta n_{\text{fixed}}$ observations of $Y_{ip}$ from $\mathbf{x}_i$ that has never been visited before; otherwise, it takes $\Delta n_{\text{free}}$ observations. NP selects the best solution over the set $\mathcal{S}_k$, $\hat{\mathbf{x}}^*(\mathcal{S}_k)$, as the one with the largest cumulative sample average, and NP uses $\hat{\mathbf{x}}^*(\mathcal{S}_k)$ to determine the next most-promising region.

**Pure random search (RS)** (Andradóttir 1996)**:** RS is a modified hill-climbing algorithm. Let $C_i(k)$ denote the number of times $\mathbf{x}_i$ becomes the current solution up to iteration $k$. RS proceeds as follows:

1. Use MIX-D to uniformly sample a candidate solution $\mathbf{x}_{I'_k}$ over $\Theta \setminus \{I_k\}$, where $I_k$ is the index of the current solution.
2. Take $\Delta n_{\text{fixed}} > 0$ observations of $Y_{I_k p}$ and $Y_{I'_k p}$, and compute the sample averages over these observations: $\bar{Y}_{I'_k}(\Delta n_{\text{fixed}})$ and $\bar{Y}_{I_k}(\Delta n_{\text{fixed}})$.
3. Update $I_k$ and $C_i(k)$:

$$\begin{aligned} I_{k+1} &= \begin{cases} I'_k, \text{ if } \bar{Y}_{I'_k}(\Delta n_{\text{fixed}}) > \bar{Y}_{I_k}(\Delta n_{\text{fixed}}) \\ I_k, \text{ otherwise} \end{cases} && (13) \\ C_i(k+1) &= C_i(k) + \mathcal{I}\{I_{k+1} = i\} \\ k &= k+1, \end{aligned}$$

where $\mathcal{I}\{v\} = 1$ if $v$ is true and 0 otherwise. The selected solution is $\mathbf{x}_{\hat{i}^*}$ where

$$\hat{i}^* \quad = \quad \text{argmax}_{1 \le i \le v} C_i(k). \qquad (14)$$

**Simulated annealing (SA)** (Andradóttir 1999)**:** SA is almost identical to RS, but occasional downhill moves are allowed. Let the neighborhood structure be such that every solution is a neighbor of every other solution. The annealing temperature $T$ is fixed. Equation (13) is replaced by

$$I_{k+1} \leftarrow \begin{cases} I'_k, \text{ if } U_k \le \exp\left\{-\Delta Y_k^+ / T\right\} \\ I_k, \text{ otherwise.} \end{cases} \qquad (15)$$

where $\Delta Y_k^+ = \max\left(\bar{Y}_{I_k}(\Delta n_{\text{fixed}}) - \bar{Y}_{I'_k}(\Delta n_{\text{fixed}}), 0\right)$, and $U_k \sim$ uniform$(0, 1)$. The selected solution is $\mathbf{x}_{\hat{i}^*}$ where

$$\hat{i}^* \quad = \quad \text{argmax}_{1 \le i \le v}\{\bar{Y}_i(n_i(k)) : n_i(k) > 0\}. \qquad (16)$$

**Remark:** The past observations are accumulated in RS and SA for the purpose of estimating the optimal solution upon search termination (in (14) and (16)), but they are not used for the local comparison (in (13) or (15)).

We consider RS and SA because they are globally convergent (Our problem setting is such that the required conditions specified in Andradóttir (1996, 1999) are satisfied.). We compare NP+SSM+HC to NP to study the role of local selection-of-the-best schemes and HC. Each optimization scheme is repeated for some number of times, and the results shown below are the *averaged* values across different searches.

### 6.1 Three-Stage Buffer Allocation Problem

We consider a three-stage flow line with finite buffer storage space in front of stations 2 and 3 and an infinite number of jobs in front of station 1. There is a single server at each station, and the service time at station $h$ is exponentially distributed with rate $\mu_h$, $h = 1, 2, 3$. If the buffer of station $h$ is full, then station $h-1$ is blocked and a finished job cannot be released from station $h-1$. The total buffer space and the service rates are limited. The goal is to find a buffer allocation and service rates such that the throughput (average output of the flow line per unit time) is maximized. We obtain the balance equations of the underlying Markov chain from Buzacott and Shantikumar (1993).

Let $b_h$ be the number of buffer space at station $h$, $h = 1, 2, 3$. The constraints (in the form of (2)) are:

$$\begin{array}{rcl}
\mu_1 + \mu_2 + \mu_3 & \leq & 20 \\
b_2 + b_3 & \leq & 20 \\
-b_2 - b_3 & \leq & -20 \\
1 \;\leq\; \mu_h & \leq & 20, \quad h = 1, 2, 3 \\
1 \;\leq\; b_h & \leq & 20, \quad h = 1, 2 \\
\mu_h, b_h & \in & \mathcal{Z}^+.
\end{array}$$

The number of feasible solution is 21,660. The optimal solutions are $(\mu_1, \mu_2, \mu_3, b_2, b_3)$ = (6, 7, 7, 12, 8) and (7, 7, 6, 8, 12) with an expected throughput of 5.776. In the simulation, the throughput is estimated after the first 2000 units have been produced, and it is averaged over the subsequent 50 units released.

All of the searches start from (2, 2, 2, 2, 18). Other parameter values for NP+SSM+HC are as follows: when NP backtracks it is to the entire feasible region $\Theta$; the partitioning criterion for NP is `BIGGEST RANGE`; the most-promising region is partitioned into $\omega = 2$ subregions; the number of solutions sampled from each subregion and the surrounding region is $\vartheta = 5$; each time a solution is visited it gets at least $\Delta n_{\text{free}} = 1$ additional observations; the minimum number of observations needed to run SSM is $n_0 = 4$; the indifference-zone parameter of SSM is $\delta = 0.5$; the confidence level for SSM is $1 - \alpha = 0.9$; the number of iterations without progress that triggers restart is $k_0 = 6$; the number of Markov-chain transitions in each application of MIX-D or MIX-DS is 10; and the annealing temperature is $T = 3$ (see (15)). When HC is used, it is performed on all iterations and stops when progress is too small (see (12)); the number of candidate solutions on each HC step is $\varpi = 3$; and the neighborhood step size is $\xi = 1$. For NP, RS and SA the number of observations per visit is $\Delta n_{\text{fixed}} = 4$. Finally, the results we report are the average over 50 independent searches.

Figure 1 shows the expected throughput of the *current* optimal-solution estimate $\mu_{\hat{i}^*}$ ($\hat{i}^*$ is defined in (5)) averaged over 50 searches at each point in time. Initially, the performance of NP+SSM(`REGION`)+HC and NP are comparable and better than other optimization methods. However, as the simulation effort increases, NP+SSM(`REGION`)+HC and NP+SSM+HC outperform the rest despite their small number of solutions seen relative to other optimization schemes (see Figure 2). This illustrates the dilemma of local selection versus global exploration. When the simulation effort is small, both NP+SSM(`REGION`)+HC and NP+SSM+HC are unable to see much of $\Theta$ because they use more simulation effort per search iteration than other schemes do (as they use SSM to select the best). However, as the optimization progresses further, the benefit of successively making good selections on every iteration finally pays off, and the performance of NP+SSM(`REGION`)+HC and NP+SSM+HC surpasses that of other optimization schemes.

The benefit of HC is manifested through the favorable performance of NP+SSM(`REGION`)+HC compared to that of NP+SSM(`REGION`). This result confirms our conjecture that HC is advantageous for problems with large $|\Theta|$ and clustered good solutions; this three-stage buffer allocation problem has $|\Theta| = 21,660$ with 5 decision variables, and good solutions are in close proximity.

Next, we examine the value of option `REGION`. NP+SSM(`REGION`)+HC noticably outperforms NP+SSM+HC in the initial phase of the search. Option `REGION` is helpful when the simulation budget is limited because SSM(`REGION`) consumes less simulation effort per search iteration than SSM does (recall that SSM(`REGION`) stops when all surviving solutions belong to the same subregion). This savings allows NP+SSM(`REGION`)+HC to explore $\Theta$ more extensively than NP+SSM+HC does (see Figure 2), and thus the search with option `REGION` is more likely to discover good solutions.

## 7 THE FUTURE

We have proposed an optimization-via-simulation algorithm with the goal of establishing both provable convergence and good empirical performance. NP+SSM+HC consists of a global guidance system, selection of the best, and local improvement. We use the NP method as our global guidance system to ensure that the search not only advances toward optimal solutions, but it also reaches one of them, if there is enough simulation effort. While the NP method gives us the convergence guarantee, SSM enhances the performance of the NP method by controlling the local-selection error, and HC improves it further by intensifying the search near good solutions.

Our motivation behind NP+SSM+HC is essentially to make optimization-via-simulation algorithms adapt to variability and to characteristics of the response surface. Our algorithms show promise in numerical tests (see the extensive study in Pichitlamken 2002). Still, we need to find methods for adapting effectively if we are to reap the full benefits of the NP+SSM+HC Algorithm.

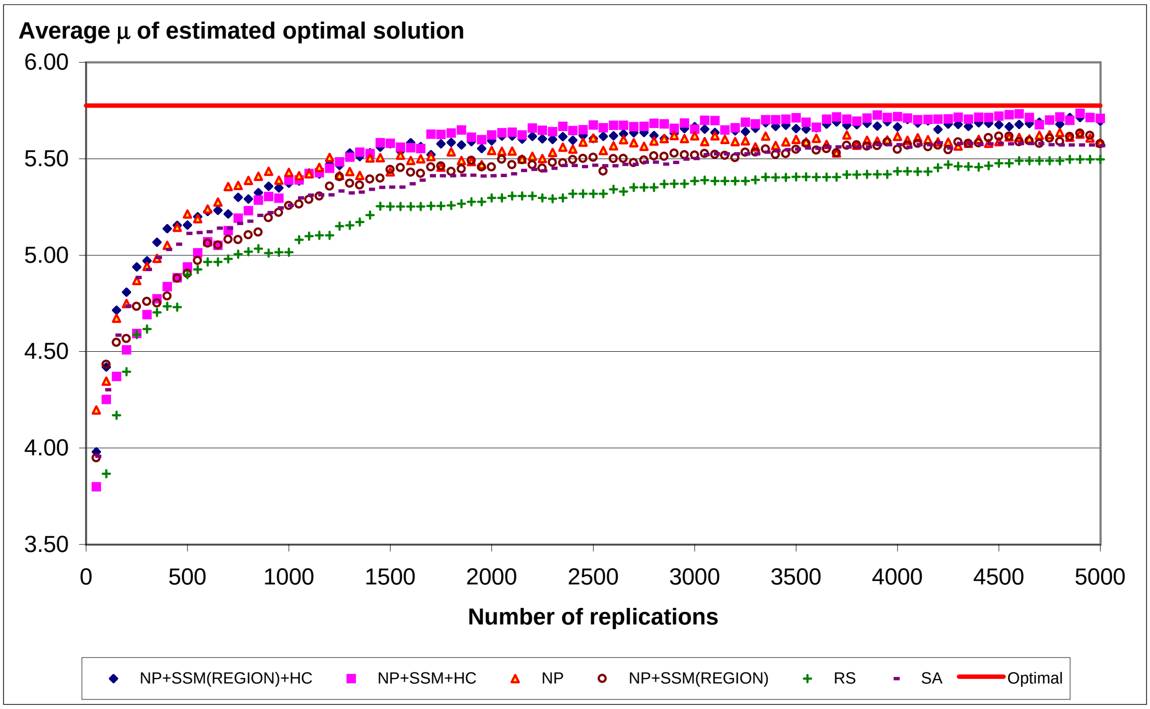

Figure 1: Expected Throughput of the Current Optimal-Solution Estimate at Each Point in Time

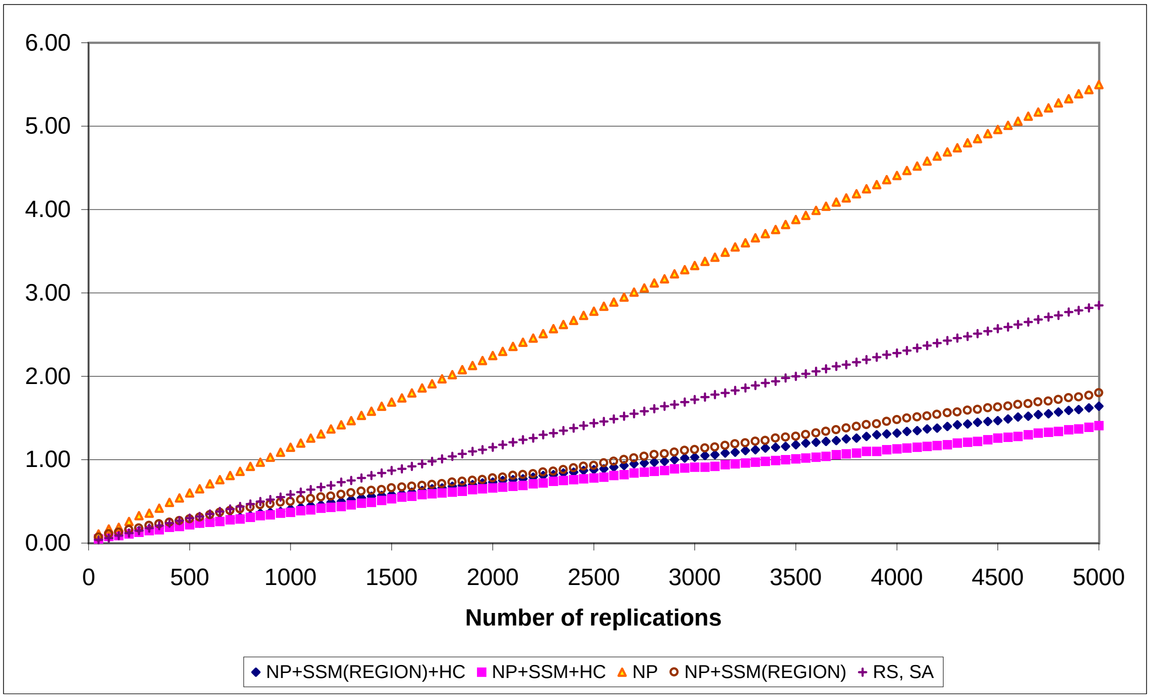

Figure 2: % of Solution Seen by the Optimization Schemes

## ACKNOWLEDGMENTS

We would like to thank General Motors R&D Center for their support of this work.

## 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. 1995. A method for discrete stochastic optimization. *Management Science* 41:1946–1961.

Andradóttir, S. 1996. A global search method for discrete stochastic optimization. *SIAM Journal on Optimization* 6:513–530.

Andradóttir, S. 1998. Simulation Optimization. *Handbook of Simulation*, ed. J. Banks, Chapter 9. New York: Wiley-Interscience.

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

Boesel, J. 1999. Search and selection for large-scale stochastic optimization. Doctoral dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

Boesel, J., B. L. Nelson, and N. Ishii. 2002. A framework for simulation-optimization software. *IIE Transactions* In press.

Buzacott, J. A., and J. G. Shantikumar. 1993. *Stochastic Models of Manufacturing Systems*. Englewood Cliffs, New Jersey: Prentice-Hall.

Gelfand, S. B., and S. K. Mitter. 1989. Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Application* 62:49–62.

Gutjahr, W. J., and G. Ch. Pflug. 1996. Simulated annealing for noisy cost functions. *Journal of Global Optimization* 8:1–13.

Pichitlamken, J. 2002. A combined procedure for optimization via simulation. Doctoral dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

Pichitlamken, J., and B. L. Nelson. 2001. Selection-of-the-best procedures for optimization via simulation. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros and M. W. Rohrer, 401–407. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via `<http://www.informs-cs.org/wsc01papers/052.PDF>` [accessed March 27, 2001].

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

Smith R. L. 1984. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research* 32:1296–1308.

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

## AUTHOR BIOGRAPHIES

**JUTA PICHITLAMKEN** received her Ph.D. from the Department of Industrial Engineering and Management Sciences at Northwestern University in 2002. She is currently an Assistant Professor in the Department of Industrial Engineering at Kasetsart University in Thailand. Her research interests include ranking and selection procedures and simulation optimization. Her e-mail address is `<juta@alumni.northwestern.edu>`.

**BARRY L. NELSON** is a Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University, and is Director of the Master of Engineering Management Program there. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems. He has published numerous papers and two books. Nelson has served the profession as the Simulation Area Editor of *Operations Research* and President of the INFORMS (then TIMS) College on Simulation. He has held many positions for the Winter Simulation Conference, including Program Chair in 1997 and current membership on the Board of Directors. His e-mail and web addresses are `<nelsonb@northwestern.edu>` and `<www.iems.northwestern.edu/˜nelsonb/>`.