A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization

Mahmoud H. Alrefaei • Sigrún Andradóttir
Department of Mathematics and Statistics, Jordan University of Science & Technology, Irbid 22110, Jordan
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

We present a modification of the simulated annealing algorithm designed for solving discrete stochastic optimization problems. Like the original simulated annealing algorithm, our method has the hill climbing feature, so it can find global optimal solutions to discrete stochastic optimization problems with many local solutions. However, our method differs from the original simulated annealing algorithm in that it uses a constant (rather than decreasing) temperature. We consider two approaches for estimating the optimal solution. The first approach uses the number of visits the algorithm makes to the different states (divided by a normalizer) to estimate the optimal solution. The second approach uses the state that has the best average estimated objective function value as estimate of the optimal solution. We show that both variants of our method are guaranteed to converge almost surely to the set of global optimal solutions, and discuss how our work applies in the discrete deterministic optimization setting. We also show how both variants can be applied for solving discrete optimization problems when the objective function values are estimated using either transient or steady-state simulation. Finally, we include some encouraging numerical results documenting the behavior of the two variants of our algorithm when applied for solving two versions of a particular discrete stochastic optimization problem, and compare their performance with that of other variants of the simulated annealing algorithm designed for solving discrete stochastic optimization problems.

(Global Optimization; Discrete Parameters; Simulated Annealing; Simulation Optimization)

1. Introduction

We consider the problem of optimizing an objective function over a discrete set of feasible solutions, in situations where the objective function values cannot be evaluated exactly and instead have to be estimated, for instance using simulation. In mathematical notation, this discrete stochastic optimization problem can be represented as:

$$\min_{x \in \mathcal{S}} f(x), \tag{1}$$

where $\mathcal{S}$ is a discrete set and $f: \mathcal{S} \to \mathbb{R}$ is a deterministic function whose values can only be estimated with some noise. Often $f(x)$ is the expected performance of a complex stochastic system using the system configuration x, so that $f(x)$ can be represented as

$$f(x) = E[h(x, Y_x)] = E[H(x)], \tag{2}$$

where h is a deterministic, real-valued function, and for all $x \in \mathcal{S}$, $H(x) = h(x, Y_x)$ and Y_x is a random variable that depends on the parameter $x \in \mathcal{S}$. Let $\mathcal{S}^*$ denote the set of global optimal solutions to the optimization problem (1)–(2).

In this paper we consider the use of simulated

0025-1909/99/4505/0748$05.00

annealing for solving discrete stochastic optimization problems. Kirkpatrick et al. (1983) and Černý (1985) introduce the idea of simulated annealing to the field of combinatorial optimization. Their procedure resembles the model of Metropolis et al. (1953) for how in the (physical) annealing process, particles of a solid arrange themselves into thermal equilibrium at a given temperature. More specifically, consider the optimization problem (1). Suppose that the exact objective function values are available and that the objective function may have multiple local optimal solutions. Simulated annealing allows "hill-climbing" moves in order to avoid local optimal solutions. This method starts with an arbitrary state in $\mathcal{S}$ as the estimate of the optimal solution. If $x \in \mathcal{S}$ is the estimated optimal solution in iteration k, then a new state x' is randomly selected from a neighborhood of x, $N(x)$ (see Definition 1 in §2). If $f(x') \leq f(x)$, so that x' is a better state than x, then the method accepts the move from x to x', and x' becomes the new estimated optimal solution. On the other hand, if $f(x') > f(x)$, then there is a chance that x' will be accepted even though it is a worse state than x; this is because good points might be "hidden behind" x'. More specifically, a uniform random number $U_k \sim U[0, 1]$ is generated (i.e., U_k is uniformly distributed on the interval [0, 1]). If $U_k \leq \exp[-[f(x') - f(x)]^+/T_k]$, where $T_k > 0$ is a controlling parameter called "temperature" and $y^+ = \max\{0, y\}$ for all $y \in \mathbb{R}$, then the state x' is accepted as the new estimate of the optimal solution; otherwise x remains the estimate of the optimal solution. Note that the higher the temperature T_k, the more likely it is that a "hill-climbing" move from x to x' with $f(x') > f(x)$ is accepted. The simulated annealing algorithm assumes that $T_k > 0$ for all k and that $T_k \to 0$ as $k \to \infty$. The initial temperature T_0 and the rate at which the sequence $\{T_k\}$ (the annealing schedule) is decreased play important roles in the convergence of the simulated annealing algorithm and the quality of the final estimate of the optimal solution.

Most authors studying simulated annealing have focused on determining an appropriate annealing schedule in order to obtain convergence in probability to the set of global optimal solutions $\mathcal{S}^*$. Hajek (1988), Theorem 1, shows that for annealing schedules $\{T_k\}$ of the form

$$T_k = \frac{C}{\ln(1+k)}, \quad \forall\, k \in \mathbb{N}, \tag{3}$$

a necessary and sufficient condition on C for the algorithm to converge in probability to the set $\mathcal{S}^*$ is that $C \geq d^*$, where d^* is the maximum depth of the local optimal solutions that are not globally optimal (see Hajek (1988), page 313, for the definition of the depth of a local optimal solution). Earlier results on the convergence of simulated annealing are given by Geman and Geman (1984) and by Mitra et al. (1986).

The original simulated annealing algorithm assumes that the objective function values can be evaluated exactly. However, in many practical problems, it is impossible to evaluate the objective function exactly, and instead the evaluation of the objective function values may include some noise. This is, for example, the case when the values of the objective function are the expected performance of a complex stochastic system under different system configurations, and are estimated using simulation (see Equation (2)). Despite the wide use of simulated annealing for solving deterministic optimization problems, until recently little effort has been spent on applying it to solve such discrete stochastic optimization problems. (However, the simulated annealing procedure can be viewed as an ordinary hill-descent method with artificial noise added; see for example Hajek and Sasaki (1989), Aluffi-Pentini et al. (1985), and Geman and Hwang (1986). See also Kushner (1963) for discussion of a random directions Kiefer-Wolfowitz stochastic approximation procedure.) We now review the literature on the use of simulated annealing for solving discrete stochastic optimization problems. For other work on random search methods for stochastic optimization, see for example Alrefaei and Andradóttir (1998), Andradóttir (1995, 1996, 1998), Devroye (1976), Gurin (1966), Marti (1982), Yakowitz and Lugosi (1990), Yan and Mukai (1992), and references therein.

As far as we are aware, the first authors to publish a simulated annealing procedure for discrete stochastic optimization are Bulgak and Sanders (1988), who propose a heuristic method and use it for solving a

buffer allocation problem. The basic idea behind their approach is to use confidence intervals to determine in each iteration whether the difference in the estimated objective function values at the current and candidate states is statistically significant.

Haddock and Mittenthal (1992) also present a heuristic simulated annealing procedure and apply it to optimize a particular manufacturing system. In their approach, whenever it is of interest to estimate the value the objective function takes at a particular state, Haddock and Mittenthal do a long simulation run to get an accurate estimated objective function value. This means that the amount of computer time required in each iteration of the algorithm is significant. In order to reduce the required computer time, they then use a more rapidly decreasing annealing schedule than the one given in Equation (3). However, the convergence of the algorithm is not guaranteed using rapidly decreasing temperatures.

The first theoretical analysis of simulated annealing applied to solve discrete stochastic optimization problems that we are aware of is given by Gelfand and Mitter (1989). They show that if the noise in the estimated objective function values in iteration k has the normal $N(0, \sigma_k^2)$ distribution with zero mean and variance $\sigma_k^2 > 0$, where $\sigma_k = o(T_k)$ as $k \to \infty$ and $T_k \to 0$ as $k \to \infty$ (the sequence $\{T_k\}$ is the annealing schedule used in the algorithm), then their procedure converges in probability to the set of global optimal solutions provided that the sequence $\{T_k\}$ is chosen properly (i.e., to guarantee the convergence in probability of the algorithm using exact objective function values; see for example Equation (3)).

Gutjahr and Pflug (1996) present convergence results for the simulated annealing algorithm studied by Gelfand and Mitter (1989). They also study the case when the noise in the estimated objective function values needed in iteration k has the normal distribution $N(0, \sigma_k^2)$. They show that when $\sigma_k = O(k^{-\gamma})$, where $\gamma > 1$, then the simulated annealing algorithm converges in probability to the set of global optimal solutions if the sequence $\{T_k\}$ is chosen properly. They also generalize their convergence proof to any noise distribution that is symmetric around zero and more peaked around zero than the $N(0, \sigma_k^2)$ distribution satisfying $\sigma_k = O(k^{-\gamma})$, where $\gamma > 1$ (i.e., the noise is a random variable X that satisfies $P\{X \in (-\epsilon, \epsilon)\} \geq P\{N(0, \sigma_k^2) \in (-\epsilon, \epsilon)\}$, for all $\epsilon > 0$). Note that in order to satisfy the variance assumptions of Gelfand and Mitter (1989) and Gutjahr and Pflug (1996), one needs to obtain more precise estimates of the objective function values as the number of iterations grows, which results in more computation time per iteration as the number of iterations becomes large.

Fox and Heine (1995) also study the use of simulated annealing to solve discrete stochastic optimization problems. Their method does not have a restrictive variance assumption. However, their method assumes that the range of the objective function values is restricted to a finite set. They also consider relaxing this assumption. Instead of restarting the simulation from scratch to estimate the objective function value $f(x)$, where $x \in \mathcal{S}$, each time an estimate of $f(x)$ is needed (as the method analyzed by Gelfand and Mitter (1989) and Gutjahr and Pflug (1996) does), they generate a few more observations from $f(x)$ and average them with the previous observations from $f(x)$ obtained in earlier iterations to obtain the desired estimate of $f(x)$. They show that this variant of simulated annealing converges in probability to the set of global optimal solutions if the sequence $\{T_k\}$ is chosen properly.

All of the methods we have discussed so far assume that the sequence $\{T_k\}$ decreases at some specific rate. An appropriate choice of the sequence $\{T_k\}$ is described in Equation (3). However, this sequence $\{T_k\}$ involves the parameter C, which is generally not known in practice. The convergence of these algorithms depends on the choice of the parameter C; i.e., if the parameter C is too small then these algorithms may end up at a local optimal solution, and if it is too large then these algorithms may take a long time to converge.

In this paper we propose a new modification of the simulated annealing algorithm designed for solving discrete stochastic optimization problems. Like the original simulated annealing algorithm, our method has the hill-climbing feature, so it can handle problems with many local optimizers. The mechanism for moving around the state space is similar to that of the

original simulated annealing algorithm. Since exact objective function values are not available, estimates of the objective function values will be used throughout. However, our method uses a different approach for estimating the optimal solution than the other simulated annealing approaches; rather than using the state that is visited by the algorithm in iteration k as the estimated optimal solution in that iteration, our method uses the state that is visited most often by the algorithm (divided by a normalizer) as the estimated optimal solution (this is similar to the approach proposed by Andradóttir (1995, 1996)).

We also consider using another approach for estimating the optimal solution and discuss how this approach is expected to accelerate the convergence of the algorithm. Using these two approaches, we do not require the Markov chain generated by our algorithm to converge to the set of global optimal solutions $\mathcal{S}^*$, so we do not have to restrict the temperature to decrease to zero. Instead, we use a constant temperature $T_k = T > 0$, for all $k \in \mathbb{N}$, to give our algorithm the freedom to move aggressively around the state space and locate an optimal solution rapidly. We show that our methods converge almost surely to the set $\mathcal{S}^*$ under mild conditions and discuss how our work applies to discrete deterministic optimization.

We also discuss how our methods can be used to solve discrete optimization problems when the objective function values are estimated using either transient or steady-state simulation. Finally, we present numerical results obtained by applying our algorithms, together with the algorithms studied by Gelfand and Mitter (1989), Gutjahr and Pflug (1996), and Fox and Heine (1995), to solve simple discrete optimization problems involving $M/M/1$ queueing systems. Parts of this paper were presented at the 1995 Winter Simulation Conference (see Alrefaei and Andradóttir 1995).

This paper is organized as follows. In §§2 and 3 we present two variants of our method and discuss their convergence. In §4 we discuss how our methods can be used to solve discrete simulation optimization problems. In §5 we compare the performance of our methods with that of other simulated annealing algorithms for discrete stochastic optimization. Finally, §6 contains some concluding remarks.

2. Simulated Annealing for Noisy Functions

In this section we present our modified simulated annealing algorithm for solving discrete stochastic optimization problems. Since exact objective function values are not available, our method uses estimates of the objective function values that become more precise over time. This makes sense since as time progresses, the current state visited by the algorithm is likely to become closer to a global optimal solution. Unlike the original simulated annealing algorithm, our algorithm uses a constant temperature and it uses the number of visits the algorithm makes to the different states (divided by a normalizer) to estimate the optimal solution; see also Andradóttir (1995, 1996). We need the following definitions and assumptions:

DEFINITION 1. For each $x \in \mathcal{S}$, there exists a subset $N(x)$ of $\mathcal{S}\backslash\{x\}$, which is called the set of neighbors of x.

ASSUMPTION 1. *For any $x, x' \in \mathcal{S}$, x' is reachable from x; i.e., there exists a finite sequence $\{n_i\}_{i=0}^{l}$ for some l, such that $x_{n_0} = x$, $x_{n_l} = x'$, and $x_{n_{i+1}} \in N(x_{n_i})$, $i = 0, 1, 2, \ldots, l-1$.*

Let $R'\colon \mathcal{S} \times \mathcal{S} \to [0, \infty)$ be a function and for all $x \in \mathcal{S}$, define $D(x) = \sum_{x' \in \mathcal{S}} R'(x, x')$ and

$$R(x, x') = \frac{R'(x, x')}{D(x)}, \tag{4}$$

for all $x' \in \mathcal{S}$. The normalization in Equation (4) allows us to replace the assumption that R be symmetric by the following more general Assumption 2 (see Equation (20) below and the subsequent discussion for an example in which R is asymmetric but Assumption 2 holds).

ASSUMPTION 2. *Let the transition probability $R(x, x')$ be defined as in Equation (4) and let N satisfy Definition 1. Then we assume that:*

1. *$R'\colon \mathcal{S} \times \mathcal{S} \to [0, \infty)$ satisfies $R'(x, x') > 0 \Leftrightarrow x' \in N(x)$, and*

2. $R'(x, x') = R'(x', x)$, *for each* $x, x' \in \mathcal{S}$.

ASSUMPTION 3. *The feasible region* $\mathcal{S}$ *is a finite set containing at least two states and the set of all global optimal solutions* $\mathcal{S}^*$ *is a proper subset of* $\mathcal{S}$.

REMARK 1. The assumption that $\mathcal{S}$ contains at least two states and that $\mathcal{S}^*$ is a proper subset of $\mathcal{S}$ can be made without loss of generality since the optimization problem (1) is trivial otherwise. Also note that Assumption 2 implies that $x' \in N(x) \Leftrightarrow x \in N(x')$.

ASSUMPTION 4. *The temperature* T *is a positive (constant) real number.*

ASSUMPTION 5. *Let* $\{L_k\}$ *be a sequence of positive integers such that* $L_k \to \infty$ *as* $k \to \infty$.

We also need the following lemma proved by Andradóttir (1995).

LEMMA 1. (ANDRADÓTTIR (1995), LEMMA 3.1). *Suppose that* $\{D_k\}$ *is a nonhomogeneous Markov chain with a finite state space* $\mathcal{S}$ *and with* $P\{D_{k+1} = d \mid D_0, \ldots, D_k\} = P_k(D_k, d)$ *for all* $d \in \mathcal{S}$ *and* $k \in \mathbb{N}$, *where* $P_k(x, x') \to P(x, x')$ *as* $k \to \infty$ *for all* $x, x' \in \mathcal{S}$ *and* P *is an irreducible and aperiodic Markov matrix. If* $g: \mathcal{S} \to \mathbb{R}$, *then*

$$(1/K)\Sigma_{k=1}^{K} g(D_k) \to \Sigma_{d \in \mathcal{S}} \pi_d g(d)$$

a.s. as $K \to \infty$, *where* π *is the steady-state distribution corresponding to* P.

Now we state the proposed simulated annealing algorithm for solving optimization problems with noisy objective function evaluations. Note that after k iterations, X_k is the current state of the Markov chain generated by the algorithm, $V_k(x)$ is the number of times the Markov chain $\{X_k\}$ has visited state x in the first k iterations, for all $x \in \mathcal{S}$, and X_k^* is the state that maximizes $V_k(x)/D(x)$, where $D(x)$ is the normalizer given in Equation (4).

Algorithm 1

Parameters: R, N, T, $\{L_k\}$.

Step 0: *Select a starting point* $X_0 \in \mathcal{S}$. *Let* $V_0(X_0) = 1$ *and* $V_0(x) = 0$, *for all* $x \in \mathcal{S}$, $x \neq X_0$. *Let* $k = 0$ *and* $X_k^* = X_0$.

Step 1: *Given* $X_k = x$, *choose* $Z_k \in N(x)$ *such that* $P[Z_k = z \mid X_k = x] = R(x, z)$ *for all* $z \in N(x)$, *where* $N(\cdot)$ *is defined in Definition* 1 *and* $R(\cdot, \cdot)$ *is given in Equation* (4).

Step 2: *Given* $X_k = x$ *and* $Z_k = z$, *generate independent, identically distributed, and unbiased observations* $Y_z(1), Y_z(2), \ldots, Y_z(L_k)$ *of* Y_z *and* $Y_x(1), Y_x(2), \ldots, Y_x(L_k)$ *of* Y_x. *Compute* $\hat{f}_{L_k}(x)$ *and* $\hat{f}_{L_k}(z)$ *as follows:*

$$\hat{f}_{L_k}(s) = \frac{1}{L_k} \sum_{i=1}^{L_k} h(s, Y_s(i)) \quad \text{for } s = x, z. \tag{5}$$

Step 3: *Given* $X_k = x$ *and* $Z_k = z$, *generate* $U_k \sim U[0, 1]$, *and set*

$$X_{k+1} = \begin{cases} Z_k & \text{if } U_k \leq G_{x,z}(k), \\ X_k & \text{otherwise,} \end{cases}$$

where

$$G_{x,z}(k) = \exp\left[\frac{-[\hat{f}_{L_k}(z) - \hat{f}_{L_k}(x)]^+}{T}\right]. \tag{6}$$

Step 4: *Let* $k = k + 1$, $V_k(X_k) = V_{k-1}(X_k) + 1$, *and* $V_k(x) = V_{k-1}(x)$, *for all* $x \in \mathcal{S}$, $x \neq X_k$. *If* $V_k(X_k)/D(X_k) > V_k(X_{k-1}^*)/D(X_{k-1}^*)$, *then let* $X_k^* = X_k$; *otherwise let* $X_k^* = X_{k-1}^*$. *Go to Step* 1.

REMARK 2. The estimates $\hat{f}_{L_k}(x)$ and $\hat{f}_{L_k}(z)$ generated in Step 2 of Algorithm 1 need not be independent of each other and can in fact be generated using common random numbers.

The stochastic process $\{X_k\}$ generated by Algorithm 1 is a time-inhomogeneous Markov chain whose transition probability matrices are given by

$$P_k(x, x') = P\{X_{k+1} = x' \mid X_k = x\}$$

$$= \begin{cases} R(x, x')P\{U_k \leq G_{x,x'}(k)\} & \text{if } x' \in N(x), \\ 1 - \Sigma_{y \in N(x)} P_k(x, y) & \text{if } x' = x, \\ 0 & \text{otherwise,} \end{cases}$$

for all $k \in \mathbb{N}$, where U_k is a uniform random variable defined on the interval [0, 1] and $G_{x,x'}(k)$ is defined in Equation (6). Note that if $x' \in N(x)$, then $P\{U_k \leq G_{x,x'}(k)\} = E[G_{x,x'}(k)]$. Therefore, the transition probability matrices P_k can be rewritten as:

$$P_k(x, x') = P\{X_{k+1} = x' \mid X_k = x\}$$

$$= \begin{cases} R(x, x')p_{x,x'}(k) & x' \in N(x), \\ 1 - \Sigma_{y \in N(x)} P_k(x, y) & \text{if } x' = x, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

for all $k \in \mathbb{N}$, where $p_{x,x'}(k) = E[G_{x,x'}(k)]$ for all $x \in \mathcal{S}$, $x' \in N(x)$, and $k \in \mathbb{N}$. Let P be the transition probability matrix defined by

$$P(x, x') = \begin{cases} R(x, x') \exp\left[\dfrac{-[f(x') - f(x)]^+}{T}\right] & \text{if } x' \in N(x), \\ 1 - \Sigma_{y \in N(x)} P(x, y) & \text{if } x' = x, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and let π_x be defined as follows:

$$\pi_x = \frac{D(x) \exp\left[\dfrac{-f(x)}{T}\right]}{\Sigma_{x' \in \mathcal{S}} D(x') \exp\left[\dfrac{-f(x')}{T}\right]} \tag{9}$$

for all $x \in \mathcal{S}$. Then we have the following two propositions.

PROPOSITION 2. *Under Assumptions* 1 *through* 3, *the transition probability matrix* P *given in Equation* (8) *is irreducible and aperiodic and has stationary distribution* π, *where* π *is a vector whose entries* π_x *are given in Equation* (9).

PROOF. The proof of irreducibility follows directly from Assumptions 1 and 2 and Equation (8). The proof that π is the stationary distribution of P can be found in Proposition 3.1 in Mitra et al. (1986). To prove aperiodicity, note that by Assumption 3, $\mathcal{S}^* \neq \mathcal{S}$, so by Assumption 1 there exist $x^* \in \mathcal{S}^*$ and $x \in N(x^*)$ with $f(x^*) < f(x)$. Therefore, from the definition of P given in Equation (8) and the definition of R given in Equation (4), $P(x^*, x^*) > 0$ and therefore, P is aperiodic. □

PROPOSITION 3. *Suppose that Assumptions* 4 *and* 5 *are satisfied and that* $\mathcal{S}$ *is finite. Then* $P_k(x, x') \to P(x, x')$ *as* $k \to \infty$ *for all* $x, x' \in \mathcal{S}$, *where the Markov matrices* P_k *and* P *are given in Equations* (7) *and* (8), *respectively.*

PROOF. If $x \in \mathcal{S}$ and $x' \in N(x)$, then

$$\begin{aligned} &\lim_{k \to \infty} P_k(x, x') \\ &= R(x, x') \lim_{k \to \infty} E\left[\exp\left[\frac{-[\hat{f}_k(x') - \hat{f}_k(x)]^+}{T}\right]\right] \\ &= R(x, x') E\left[\lim_{k \to \infty} \exp\left[\frac{-[\hat{f}_k(x') - \hat{f}_k(x)]^+}{T}\right]\right] \\ &= R(x, x') E\left[\exp\left[\frac{-[f(x') - f(x)]^+}{T}\right]\right] = P(x, x'), \end{aligned}$$

where the second equality follows from the bounded convergence theorem and the third equality holds by the strong law of large numbers and Assumption 5. By Equations (7) and (8), this implies that $\lim_{k \to \infty} P_k(x, x') = P(x, x')$ for all $x, x' \in \mathcal{S}$. □

Now we have the following convergence theorem.

THEOREM 4. *Under Assumptions* 1 *through* 5, *the sequence* $\{X_k^*\}$ *generated by Algorithm* 1 *converges almost surely to the set* $\mathcal{S}^*$ (*in the sense that there exists a set* A *such that* $P(A) = 1$ *and for all* $\omega \in A$, *there exists* $K_\omega > 0$ *such that* $X_k^*(\omega) \in \mathcal{S}^*$ *for all* $k \geq K_\omega$).

PROOF. It is clear that the sequence $\{X_k\}$ generated by Algorithm 1 is a time-inhomogeneous Markov chain with transition probability matrices P_k defined in Equation (7). By Proposition 3, $P_k \to P$ as $k \to \infty$. By Proposition 2, the matrix P is irreducible and aperiodic and has the stationary distribution π, where π is a vector whose entries π_x are given in Equation (9). By Lemma 1 (let $D_k = X_k$ for all $k \in \mathbb{N}$ and $g(d) = I_{\{d=x\}}$ for all $d \in \mathcal{S}$), we have that

$$\frac{V_k(x)}{kD(x)} = \frac{1}{D(x)} \times \frac{1}{k} \sum_{i=0}^{k} I_{\{X_i = x\}} \to \frac{\pi_x}{D(x)} \quad \text{a.s. as } k \to \infty$$

for all $x \in \mathcal{S}$, where $V_k(x)$, $x \in \mathcal{S}$, is defined before the statement of Algorithm 1. It is clear that $\pi_x > 0$ for all $x \in \mathcal{S}$ so that Equation (9) yields

$$\frac{\pi_y / D(y)}{\pi_x / D(x)} = \exp\left[\frac{-[f(y) - f(x)]}{T}\right]$$

for all $x, y \in \mathcal{S}$. Therefore, $\pi_y / D(y) \leq \pi_x / D(x)$ if and only if $f(y) \geq f(x)$. This shows that

$$\arg\max_{x\in\mathcal{S}} \frac{\pi_x}{D(x)} = \mathcal{S}^*. \tag{10}$$

Now let A be such that $P(A) = 1$, and for all $\omega \in A$, $V_k(x, \omega)/k \to \pi_x$ as $k \to \infty$, for all $x \in \mathcal{S}$. Let $\omega \in A$. Then, since $\mathcal{S}$ is finite and since $V_k(x, \omega)/kD(x) \to \pi_x/D(x)$ as $k \to \infty$, for all $x \in \mathcal{S}$, it follows from Equation (10) that there exists K_ω such that for all $k \geq K_\omega$, $V_k(x^*, \omega)/D(x^*) > V_k(x, \omega)/D(x)$ for all $k \geq K_\omega$, $x^* \in \mathcal{S}^*$, and $x \in \mathcal{S}\backslash\mathcal{S}^*$. Hence $X_k^*(\omega) \in \mathcal{S}^*$ for all $k \geq K_\omega$. □

REMARK 3. Note that Algorithm 1 converges not only with estimates of the objective function values of the form given in Equation (5), but also with any strongly consistent estimates of the objective function values.

REMARK 4. For a discrete deterministic optimization problem, Theorem 4 shows that Algorithm 1 using exact objective function values instead of estimated objective function values in Steps 2 and 3 of the algorithm converges almost surely to the set of global optimal solutions under Assumptions 1, 2, 3, and 4.

3. Another Variant of the Modified Simulated Annealing Algorithm

In this section we present another variant of the modified simulated annealing algorithm (Algorithm 1). This variant uses the same mechanism for moving around the state space as Algorithm 1, but a different approach for estimating the optimal solution. This approach selects the state with the best average estimated objective function value obtained from all the previous estimates of the objective function values to be the estimated optimal solution; this approach for estimating the optimal solution was proposed by Andradóttir (1998). Before we state the algorithm we need the following assumption:

ASSUMPTION 5′. *$\{L_k\}$ is a sequence of positive integers satisfying* $\lim_{k\to\infty} L_k = L \leq \infty$.

Algorithm 2

Parameters: R, N, T, $\{L_k\}$.

Step 0: *Select a starting point* $X_0 \in \mathcal{S}$. *For all* $x \in \mathcal{S}$, *let* $A_0(x) = 0$ *and* $C_0(x) = 0$. *Let* $k = 0$ *and* $X_k^* = X_0$.

Step 1: *Identical to Step* 1 *of Algorithm* 1.

Step 2: *Given* $X_k = x$ *and* $Z_k = z$, *generate independent, identically distributed, and unbiased observations* $Y_z(1), Y_z(2), \ldots, Y_z(L_k)$ *of* Y_z *and* $Y_x(1), Y_x(2), \ldots, Y_x(L_k)$ *of* Y_x. *Compute* $\hat{f}_{L_k}(x)$ *and* $\hat{f}_{L_k}(z)$ *as in Equation* (5) *for* $s = x, z$. *Let* $C_{k+1}(s) = C_k(s) + L_k$ *and* $A_{k+1}(s) = A_k(s) + L_k\hat{f}_{L_k}(s)$ *for* $s = x, z$. *Moreover, let* $C_{k+1}(x') = C_k(x')$ *and* $A_{k+1}(x') = A_k(x')$ *for all* $x' \in \mathcal{S}$, $x' \neq x, z$.

Step 3: *Identical to Step* 3 *of Algorithm* 1.

Step 4: *Let* $k = k + 1$ *and select* $X_k^* \in \arg\min_{x\in\mathcal{S}} A_k(x)/C_k(x)$ (*use the convention* $0/0 = +\infty$). *Go to Step* 1.

The stochastic process $\{X_k\}$ generated by Algorithm 2 is a time-inhomogeneous Markov chain whose transition probability matrices P_k are given in Equation (7). We have the following convergence theorem.

THEOREM 5. *Suppose that Assumptions* 1, 2, 3, 4, *and* 5′ *are satisfied. Then the sequence* $\{X_k^*\}$ *generated by Algorithm* 2 *converges almost surely to the set* $\mathcal{S}^*$ (*in the sense that there exists a set* A *such that* $P\{A\} = 1$ *and for all* $\omega \in A$, *there exists* $K_\omega > 0$ *such that* $X_k^*(\omega) \in \mathcal{S}^*$ *for all* $k \geq K_\omega$).

PROOF. We first prove the theorem for the case $L = \infty$. In this case, the Markov chain $\{X_k\}$ generated by Algorithm 2 coincides with the Markov chain generated by Algorithm 1. By Proposition 3, $P_k \to P$ as $k \to \infty$ where the transition probability matrices P_k and P are given in Equations (7) and (8), respectively. By Proposition 2, the transition probability matrix P is irreducible and aperiodic and has the stationary distribution π, where π is a vector whose entries π_x are given in Equation (9). Moreover, since P is irreducible and $\mathcal{S}$ is finite, we have that $\pi_x > 0$ for all $x \in \mathcal{S}$. Therefore, by Lemma 1, $V_k(x)/k \to \pi_x > 0$ a.s. as $k \to \infty$ for all $x \in \mathcal{S}$, where $V_k(x)$ is the number of visits the Markov chain $\{X_k\}$ makes to state x in the first k iterations for all $x \in \mathcal{S}$. This implies that $V_k(x) \to \infty$ a.s. as $k \to \infty$ for all $x \in \mathcal{S}$. Clearly $C_k(x) \geq V_k(x) - 1$ for all $x \in \mathcal{S}$ and $k \in \mathbb{N}$, where $C_k(x)$ is the number of times an estimate of $f(x)$ has been obtained in the first k iterations for all $x \in \mathcal{S}$. Therefore, by the strong law of large numbers, we have that $A_k(x)/C_k(x) \to f(x)$ a.s. as $k \to \infty$ for all $x \in \mathcal{S}$. This proves the theorem for the case $L = \infty$ (see also the last paragraph of the proof of Theorem 4).

If $L < \infty$ then it is easy to show that $P_k(x, x') \to P'(x, x')$ as $k \to \infty$ for all $x, x' \in \mathcal{S}$, where

$$P'(x, x') = P\{X_{k+1} = x' \mid X_k = x\}$$

$$= \begin{cases} R(x, x')p'_{x,x'} & x' \in N(x), \\ 1 - \sum_{y \in N(x)} P'(x, y) & \text{if } x' = x, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

$p'_{x,x'} = E[\exp[-[\hat{f}(x') - \hat{f}(x)]^+/T]]$, and $\hat{f}(s) = (1/L)\sum_{i=1}^{L} h(s, Y_s(i))$ for $s = x, x'$.

By Jensen's inequality, we have that

$$p'_{x,x'} \geq \exp\left[\frac{-1}{T} E[\hat{f}(x') - \hat{f}(x)]^+\right]$$

$$\geq \exp\left[\frac{-1}{T} E[\,|\hat{f}(x')| + |\hat{f}(x)|\,]\right]$$

$$> 0,$$

because $E[\,|\hat{f}(s)|\,] \leq E[\,|h(s, Y_s)|\,] < \infty$ for all $s \in \mathcal{S}$ since $f(s) = E[h(s, Y_s)] \in \mathbb{R}$ for all $s \in \mathcal{S}$. (From real analysis, recall that $E[X] = E[X^+] - E[X^-] \in \mathbb{R}$ implies $E[X^+] < \infty$ and $E[X^-] < \infty$, and therefore $E[\,|X|\,] = E[X^+] + E[X^-] < \infty$.) Therefore, by Assumptions 1 and 2, P' is irreducible. Therefore, since $\mathcal{S}$ is finite, P' has a stationary distribution, say r, with $r_x > 0$ for all $x \in \mathcal{S}$.

To prove that P' is aperiodic, we will show that there exists $x \in \mathcal{S}$ such that $P'(x, x) > 0$. Suppose that $P'(x, x) = 0$ for all $x \in \mathcal{S}$. By Assumptions 1 and 2, Equation (11), and the fact that $\exp[-[\hat{f}(x') - \hat{f}(x)]^+/T] \leq 1$, $\forall x, x' \in \mathcal{S}$, we have

$$P'(x, x) = 0, \ \forall x \in \mathcal{S}$$

$$\Leftrightarrow \sum_{x' \in N(x)} P'(x, x') = 1, \ \forall x \in \mathcal{S}$$

$$\Leftrightarrow E\left[\exp\left[\frac{-[\hat{f}(x') - \hat{f}(x)]^+}{T}\right]\right] = 1,$$

$$\forall x \in \mathcal{S}, x' \in N(x)$$

$$\Leftrightarrow \exp\left[\frac{-[\hat{f}(x') - \hat{f}(x)]^+}{T}\right] = 1 \text{ a.s.},$$

$$\forall x \in \mathcal{S}, x' \in N(x)$$

$$\Leftrightarrow \hat{f}(x') - \hat{f}(x) \leq 0 \text{ a.s., } \forall x \in \mathcal{S}, x' \in N(x)$$

$$\Leftrightarrow \hat{f}(x') = \hat{f}(x) \text{ a.s., } \forall x \in \mathcal{S}, x' \in N(x)$$

$$\Leftrightarrow \hat{f}(x') = \hat{f}(x) \text{ a.s., } \forall x, x' \in \mathcal{S}$$

$$\Leftrightarrow f(x') = f(x), \ \forall x, x' \in \mathcal{S}.$$

Since this contradicts Assumption 3, we have shown that P' is aperiodic.

Since $P_k \to P'$ as $k \to \infty$, where P' is irreducible and aperiodic and P' has a stationary distribution r, with $r_x > 0$ for all $x \in \mathcal{S}$, by Lemma 1, $V_k(x)/k \to r_x > 0$ a.s. as $k \to \infty$ for all $x \in \mathcal{S}$. This implies that $V_k(x) \to \infty$ a.s. as $k \to \infty$ for all $x \in \mathcal{S}$. The rest of the proof is similar to the proof for the case $L = \infty$. □

REMARK 5. If L is large enough to guarantee that the estimated objective function values $\hat{f}_k(x)$ are "precise" estimates of $f(x)$ for large enough values of k and all $x \in \mathcal{S}$ (for example if $L = \infty$), then the Markov chain $\{X_k\}$ generated by Algorithm 2 is attracted by good states rather than bad states; i.e., it tends to spend more time at the good states (see Theorem 4). Therefore, good estimates of the objective function values at the good states will be obtained quickly. Thus, it is reasonable to expect that Algorithm 2 will converge to the set of global optimal solutions to the optimization problem (1) more rapidly than Algorithm 1. In practice, it is generally difficult to determine a priori how large L needs to be for attraction to the good states to hold. If L is not chosen optimally, then the almost sure convergence of Algorithm 2 is still guaranteed, but the (empirical) convergence rate of the algorithm will be suboptimal. In §5, we will see how the use of Algorithm 2 improves the (empirical) convergence rate relative to Algorithm 1 when applied to solve two versions of a particular discrete stochastic optimization problem.

REMARK 6. Note that in the discrete deterministic optimization case where the objective function values are evaluated exactly, Algorithm 2 selects the state in $\mathcal{S}$ that has the best objective function value among the states that have been visited by the algorithm so far to be the estimate of the optimal solution. Theorem 5 shows that Algorithm 2, using exact objective function values instead of estimated objective function values, converges almost surely to the set of global optimal solutions under Assumptions 1, 2, 3, and 4. By Theorem 4, Algorithm 2 is attracted to the good states when L is "large enough." Therefore, Algorithm 2 is expected to obtain a good estimate of the optimal solution quickly in this setting.

4. Discrete Optimization in Simulation

In this section, we discuss how Algorithms 1 and 2 can be applied to solve discrete optimization problems in both transient and steady-state simulation. In the transient case, we are interested in solving the following optimization problem:

$$\min_{x\in\mathcal{S}} f^{(t)}(x) = E[h(x, Y_0(x), \ldots, Y_{\tau_x}(x))], \quad (12)$$

where for all $x \in \mathcal{S}$, $\{Y_i(x)\}$ is a stochastic process whose distribution depends on x and τ_x is a stopping time with respect to $\{Y_i(x)\}$. We assume that it is possible to generate independent samples $Y_0^j(x), \ldots, Y_{\tau_x^j}^j(x)$, $j = 1, 2, \ldots$, from $Y_0(x), \ldots, Y_{\tau_x}(x)$ for all $x \in \mathcal{S}$. Then it is straightforward to apply Algorithms 1 and 2 to solve the discrete optimization problem (12). If an estimate of $f^{(t)}(x)$ is needed in iteration k of these algorithms, where $x \in \mathcal{S}$ and $k \in \mathbb{N}$, then we generate L_k independent sample paths $Y_0^j(x), \ldots, Y_{\tau_x^j}^j(x)$, $j = 1, \ldots, L_k$, where the sequence $\{L_k\}$ satisfies either Assumption 5 or Assumption 5′ based on which of Algorithms 1 or 2 is being used, and take $\hat{f}_k^{(t)}(x)$ as the desired estimate of $f^{(t)}(x)$, where

$$\hat{f}_k^{(t)}(x) = \frac{1}{L_k}\sum_{j=1}^{L_k} h(s, Y_0^j(x), \ldots, Y_{\tau_x^j}^j(x)). \quad (13)$$

The convergence results presented in the previous sections can be used to prove the convergence of Algorithms 1 and 2 when they are applied to solve the optimization problem (12) using estimated objective function values of the form given in Equation (13).

We now discuss the convergence of Algorithms 1 and 2 when they are applied to solve discrete optimization problems in steady-state simulation. Let $\{Y_i(x)\}$ be a stochastic process whose distribution depends on the parameter x for all $x \in \mathcal{S}$. Suppose that for all $x \in \mathcal{S}$, there exists $f^{(s)}(x) \in \mathbb{R}$ such that

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} h(x, Y_i(x)) = f^{(s)}(x) \text{ a.s.} \quad (14)$$

Then we are interested in solving the following optimization problem:

$$\min_{x\in\mathcal{S}} f^{(s)}(x), \quad (15)$$

where $f^{(s)}(x)$ is the limit given in Equation (14) for all $x \in \mathcal{S}$.

Suppose that it is possible to generate sample paths of the stochastic process $\{Y_i(x)\}$ through simulation. To apply Algorithms 1 and 2, we need strongly consistent estimates of the objective function values. We accomplish this as follows. To get an estimate of $f^{(s)}(x)$ in iteration k, we generate a sample path $Y_1(x), \ldots, Y_{L_k}(x)$, where the sequence $\{L_k\}$ satisfies Assumption 5, and take $\hat{f}_k^{(s)}(x)$ to be the desired estimate of $f^{(s)}(x)$, where

$$\hat{f}_k^{(s)}(x) = \frac{1}{L_k}\sum_{i=1}^{L_k} h(x, Y_x(i)). \quad (16)$$

REMARK 7. If $\{Y_t(x)\}$ is a continuous time stochastic process, then we assume that $\{L_k\}$ is a real-valued sequence with $L_k > 0$ for all $k \in \mathbb{N}$ and $L_k \to \infty$ as $k \to \infty$. For all $x \in \mathcal{S}$ and $k \in \mathbb{N}$, let

$$\hat{f}_k^{(s)}(x) = \frac{1}{L_k}\int_0^{L_k} h(x, Y_t(x))dt.$$

If it is of interest to solve the optimization problem (15), where $\lim_{k\to\infty} \hat{f}_k^{(s)}(x) = f^{(s)}(x)$ a.s. for all $x \in \mathcal{S}$, then let $\hat{f}_k^{(s)}(x)$ be the estimate of $f^{(s)}(x)$ used in iteration k of Algorithms 1 and 2 (if needed).

The proof of the following theorem resembles the proofs of Theorems 4 and 5 (see also Remark 3).

THEOREM 6. *If Assumptions* 1 *through* 5 *are satisfied, if the estimates* $\hat{f}_k^{(s)}(x)$ *given in Equation* (16) *are independent of each other for all* $x \in \mathcal{S}$ *and* $k \in \mathbb{N}$, *and if Equation* (14) *holds for all* $x \in \mathcal{S}$, *then the sequences* $\{X_k^*\}$ *generated by Algorithms* 1 *and* 2 *using the estimators* $\hat{f}_k$ *defined in Equation* (16) *in Step* 2 *of the algorithm, converge almost surely to the set* $\mathcal{S}^*$ *(in the sense that there exists a set* A *such that* $P\{A\} = 1$ *and for all* $\omega \in A$, *there exists* $K_\omega > 0$ *such that* $X_k^*(\omega) \in \mathcal{S}^*$ *for all* $k \geq K_\omega$*).*

REMARK 8. Note that it is not difficult to show that Equation (14) is satisfied; see for example Theorem 2.2 in Chapter 2 of Shedler (1993).

5. Numerical Application

In this section we present numerical results obtained by applying Algorithms 1 and 2 to solve discrete optimization problems involving $M/M/1$ queueing systems and compare their performance with that of the other two variants of the simulated annealing algorithm designed for solving discrete stochastic optimization problems. One of these is the method proposed and analyzed by Gelfand and Mitter (1989) (see §3 of their paper) and by Gutjahr and Pflug (1996) (which we will refer to as the GM-GP method). The other is the method proposed and analyzed by Fox and Heine (1995) (which we will refer to as the FH method). These variants of the simulated annealing algorithm use a decreasing annealing schedule $\{T_k\}$. Like our algorithms, they use estimated objective function values since the exact objective function values are not available. Algorithms 1 and 2 as well as the GM-GP algorithm use independent estimated objective function values in the different iterations (i.e., every time an estimate of the objective function value $f(x)$ is needed, a new simulation is started from scratch). The FH algorithm does not require independent estimates of the objective function values in the different iterations. Instead, each time an estimate of the objective function value $f(x)$, where $x \in \mathcal{S}$, is needed, the algorithm generates a few more observations from $f(x)$ and averages them with the previous observations to get the desired estimate of $f(x)$. We will apply the GM-GP and FH algorithms using a sequence $\{T_k\}$ of the form given in Equation (3).

Consider an $M/M/1$ queueing system with arrival rate λ and service rate μ. For $i \in \mathbb{N}$, let W_i be the system time of customer i, S_i be the service time of customer i, and A_i be the inter-arrival time between customers $i-1$ and i, where $A_0 = 0$ and $W_0 = 0$. To generate the system time of customer c, W_c, one can generate S_i, A_i for $i = 1, \ldots, c$ and then use the following well known recursive formula:

$$W_i = \max\{S_i, W_{i-1} + S_i - A_i\} \tag{17}$$

to obtain W_i for $i = 1, \ldots, c$. We are interested in solving the following optimization problem:

$$\min_{x \in \mathcal{S}} f(x) = E[W(x)], \tag{18}$$

where $\mathcal{S} = \{1, \ldots, 50\}$ and $W(x)$ is the system time per customer in an $M/M/1$ queueing system with fixed arrival rate $\lambda = 1$ and service rate $\mu(x) > 1$ for all $x \in \mathcal{S}$, where the values of $\mu(x)$ for all $x \in \mathcal{S}$ are given in Table 1.

We solve this problem in two different settings. In the first setting, we are interested in minimizing the expected steady-state system time per customer, $f^{(s)}(x) = E[W^{(s)}(x)]$, where $x \in \mathcal{S}$. It is well known that the expected steady-state system time per customer is $f^{(s)}(x) = 1/(\mu(x) - \lambda)$ for all $x \in \mathcal{S}$. In the second setting, we are interested in minimizing the expected average system time per customer for the first 100 customers, $f^{(t)}(x) = E[\sum_{i=1}^{100} W_i(x)/100]$, where $x \in \mathcal{S}$. These two versions of the optimization problem (18) have also been considered by Alrefaei and Andradóttir (1998). Figure 1 shows the graphs of $f^{(s)}(x)$ and $\hat{f}^{(t)}(x)$, where $x \in \mathcal{S}$ and the $\hat{f}^{(t)}(x)$ are estimates of $f^{(t)}(x)$ for all $x \in \mathcal{S}$ obtained from separate long simulation runs yielding narrow 95% confidence intervals. It is clear that the estimates $\hat{f}^{(t)}(x)$ of $f^{(t)}(x)$ are consistently lower than $f^{(s)}(x)$ for all $x \in \mathcal{S}$. This is because $f^{(t)}$ is lower than $f^{(s)}$, not because there is any bias in the estimates $\hat{f}^{(t)}$.

We apply Algorithms 1 and 2 to solve the transient and steady-state versions of the optimization problem (18) using several choices of the parameter T, the neighborhood structure $\{N(\cdot)\}$, and the transition probabilities $\{R(\cdot, \cdot)\}$, to see their effect on the behavior of the algorithms. In particular, we take $T \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$ and we use two different neighborhood structures $\{N(x): x \in \mathcal{S}\}$. The first neighborhood structure is given by

$$N_1(x) = \mathcal{S} \backslash \{x\}, \quad \forall\, x \in \mathcal{S}; \tag{19}$$

when this neighborhood structure is used, we take $R_1(x, x') = R_1'(x, x') = 1/(|\mathcal{S}| - 1)$, $\forall x, x' \in \mathcal{S}$, $x \neq x'$ (so that $D(x) = 1$ for all $x \in \mathcal{S}$). This neighborhood structure corresponds essentially to pure random search. The second neighborhood structure is given by

$$N_2(x) = \begin{cases} \{2\} & \text{if } x = 1, \\ \{x \pm 1\} & \text{if } 2 \leq x \leq 49, \\ \{49\} & \text{if } x = 50. \end{cases} \tag{20}$$

For this neighborhood structure, we let $R_2'(x, x') = 1$

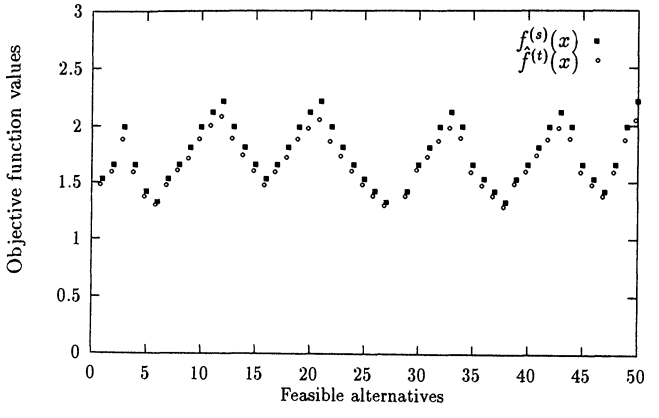
Table 1 **The Values of the Service Rates $\mu(x)$ for All $x \in \mathcal{S}$**

$\mu(1), \ldots, \mu(10)$	1.65	1.6	1.5	1.6	1.7	1.75	1.65	1.6	1.55	1.5
$\mu(11), \ldots, \mu(20)$	1.47	1.45	1.5	1.55	1.6	1.65	1.6	1.55	1.5	1.47
$\mu(21), \ldots, \mu(30)$	1.45	1.5	1.55	1.6	1.65	1.7	1.75	2.0	1.7	1.6
$\mu(31), \ldots, \mu(40)$	1.55	1.5	1.47	1.5	1.6	1.65	1.7	1.75	1.65	1.6
$\mu(41), \ldots, \mu(50)$	1.55	1.5	1.47	1.5	1.6	1.65	1.7	1.6	1.5	1.45

and $D_2(x) = |N_2(x)|$, $\forall x \in \mathcal{S}$, $x' \in N_2(x)$ (so that $R_2(x, x') = 1/|N_2(x)|$, $\forall x \in \mathcal{S}$, $x' \in N_2(x)$). Note that for the first neighborhood structure N_1, there is only one global optimal solution at $x = 28$ and no local solutions, whereas for the second neighborhood structure N_2 there are five different local solutions and one global optimal solution at $x = 28$. In the steady-state setting, the optimal value of the objective function is 1.0, whereas in the transient setting, the estimated optimal objective function value is 0.9790. We always select the initial state X_0 of Algorithms 1 and 2 randomly (uniformly) over $\mathcal{S}$ and we run 100 replications to estimate the average performance of Algorithms 1 and 2 in these two different settings.

In the transient setting, to obtain an estimate of the objective function value $f^{(t)}(x)$ in iteration k (if needed), we generate independent service times $S_1^j(x), \ldots, S_{100}^j(x)$ and inter-arrival times $A_1^j, \ldots, A_{100}^j$, for all $j = 1, \ldots, L_k$, where the sequence $\{L_k\}$ satisfies Assumption 5 for Algorithm 1 and Assumption 5′ for Algorithm 2. We use a slowly increasing sequence $\{L_k\}$ for Algorithm 1; i.e., $L_k = \lfloor \ln(10 + k) \rfloor$ for all $k \in \mathbb{N}$,

Figure 1 **The Values of the Two Objective Functions $f^{(s)}(x)$ and $\hat{f}^{(t)}(x)$ for All $x \in \mathcal{S}$**



where $\lfloor x \rfloor$ denotes the integer part of $x \in \mathbb{R}$. For Algorithm 2, we let $L_k = 1$ for all $k \in \mathbb{N}$. Then we use the recursive formula given in Equation (17) to generate $W_1^j(x), \ldots, W_{100}^j(x)$ and let

$$\hat{f}_k^{(t)}(x) = \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{1}{100} \sum_{i=1}^{100} W_i^j(x)$$

be the desired estimate of $f^{(t)}(x)$. In the steady-state setting, to obtain an estimate of the objective function value $f^{(s)}(x)$ in iteration k (if needed), we generate independent service times $S_1(x), \ldots, S_{L_k}(x)$ and interarrival times $A_1, \ldots, A_{L_k}$, where the sequence $\{L_k\}$ satisfies Assumption 5. Then we use the recursive formula of Equation (17) to generate $W_1(x), \ldots, W_{L_k}(x)$ and let $\hat{f}_k^{(s)}(x) = \sum_{i=1}^{L_k} W_i(x)/L_k$ be the desired estimate of $f^{(s)}(x)$. In both settings (transient and steady-state), we generate (conditionally) independent estimates of $f(X_k)$ and $f(Z_k)$ when Step 2 is executed in iteration k of Algorithms 1 and 2.

It is clear that the conditions of Theorems 4 and 5 are satisfied for the parameters used in this section. It is also clear that since for all $x \in \mathcal{S}$, $\lambda/\mu(x) < 1$, we have that the Markov chain $\{W_i(x)\}$ is a regenerative process with regeneration occurring each time $W_i(x) = S_i(x)$ for all $x \in \mathcal{S}$. This can be shown to imply that Equation (14) holds for all $x \in \mathcal{S}$ (see for example Corollaries III.7.5 and V.1.5 and Theorem V.3.1 in Asmussen (1987), and Theorem 9 in Chapter 9 of Wolff (1989)). Therefore, the conditions of Theorem 6 are satisfied also, so that Algorithms 1 and 2 are guaranteed to converge almost surely to the set of global optimal solutions for both the transient and steady-state versions of the optimization problem (18).

Figure 2 shows the performance of Algorithm 1 as a function of the parameter T when it is applied to solve the transient version of the optimization problem (18) using the neighborhood structure N_1 given in

Equation (19). The x-axis shows the average number of served customers that were used in our simulation, while the y-axis shows the average estimated optimal objective function value at the estimated optimal solution over the 100 replications that we performed. It is clear from this figure that in this setting choosing $T = 0.1$, 0.01, or 0.001 gives better performance than the other two choices of T ($T = 0.5$ or 1.0).

Figure 3 shows the performance of Algorithm 2 as a function of the parameter T when it is applied to solve the transient version of the optimization problem (18) using the neighborhood structure N_1 given in Equation (19). It is clear from this figure that all choices of the parameter T presented yield good performance of Algorithm 2. This suggests that Algorithm 2 is not very sensitive to the choice of the parameter T.

Figure 4 shows the performance of Algorithm 1 as a function of the parameter T when it is applied to solve the transient version of the optimization problem (18) using the neighborhood structure N_2 given in Equation (20). As in Figure 2, Figure 4 indicates that choosing $T = 0.1$, 0.01, or 0.001 gives a better performance of Algorithm 1 in this setting than larger values of T. We believe that this is because when a large value of T is used, Algorithm 1 moves around the state space very aggressively, so it may take a long time before the number of visits to the good states becomes larger than the number of visits to the bad states. From Figure 4, we conclude that the neighborhood structure N_1 yields a better performance of Algorithm 1 than the neighborhood structure N_2 and that the algorithm does not behave well using the smaller neighborhood structure N_2. This is because with the neighborhood structure N_2 there are many local solutions, whereas with the neighborhood structure N_1 there is only one global optimal solution and no other local solutions.

Figure 5 shows the performance of Algorithm 2 as a function of the parameter T when it is applied to solve the transient version of the optimization problem (18) using the neighborhood structure N_2 given in Equation (20). Figure 5 indicates that the neighborhood structure N_1 also gives a better performance of Algorithm 2 than the neighborhood structure N_2. Moreover, Figure 5 confirms the conclusion drawn from Figure 3 that the choice of the parameter T does not

Figure 2 **Performance of Algorithm 1 in the Transient Setting as a Function of the Parameter T Using the Neighborhood Structure N_1**

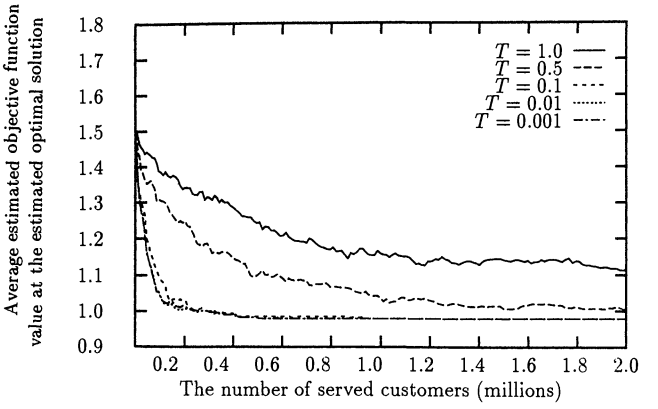
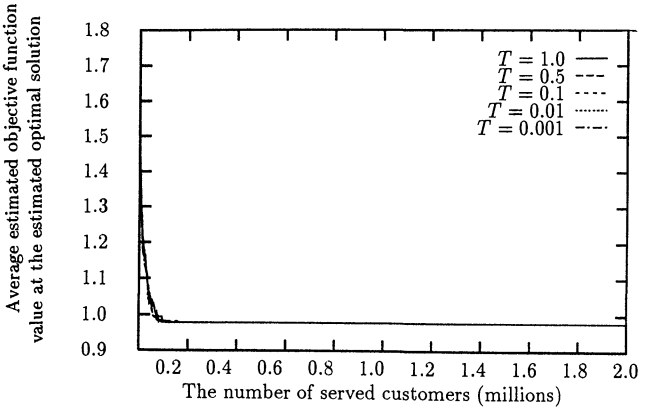


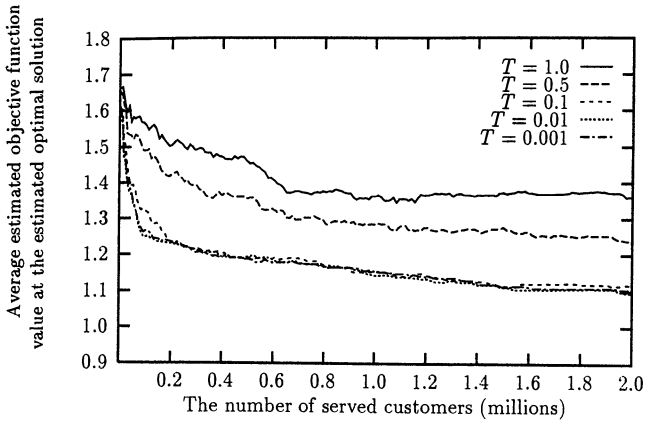
Figure 3 **Performance of Algorithm 2 in the Transient Setting as a Function of the Parameter T Using the Neighborhood Structure N_1**



appear to have a large influence on the behavior of Algorithm 2. From Figures 2, 3, 4, and 5, we conclude that for this particular problem, it is better to use a large neighborhood structure because this decreases the number of local optimal solutions.

Tables 2 and 3 compare the performance of Algorithms 1 and 2 with that of the GM-GP and FH algorithms when they are applied to solve the transient version of the optimization problem (18) using the following choices of the parameter T, the annealing sequence $\{T_k\}$ (for the GM-GP and FH algorithms), and the sequence $\{L_k\}$: $T = 0.01$, and for all $k \in \mathbb{N}$, $T_k = 0.1/\ln(10 + k)$, $L_k = \lfloor \ln(10 + k) \rfloor$ for Algorithm 1, $L_k = 1$ for Algorithm 2, $L_k = 1 + \lfloor k/20 \rfloor$ for the GM-GP

Figure 4 **Performance of Algorithm 1 in the Transient Setting as a Function of the Parameter T Using the Neighborhood Structure N_2**



algorithm, and $L_k = 1$ for the FH algorithm. Table 2 shows the results using the neighborhood structure N_1 described in Equation (19). Column 1 of Table 2 shows the number of iterations these algorithms have done so far. Column 2 shows the number of convergent replications of Algorithm 1 out of a total of 100 replications (i.e., the number of replications in which the estimated optimal solution equals the (true) global optimal solution). Column 3 shows the average number of customers used to estimate the objective function values needed by Algorithm 1 so far. This column estimates the computation effort expended so far by the algorithm. The next three sets of two columns each show the same information about the performance of Algorithm 2 and the GM-GP and FH algorithms, respectively. It is clear from this table that Algorithm 2 has the best performance in terms of the number of customers used to estimate the required objective function values in this setting. The FH algorithm shows the second best performance in this setting. Finally, Algorithm 1 performs better than the GM-GP algorithm in this setting. (This ranking of the algorithms is based on how many replications have converged as a function of the estimated computational effort expended. For example, after an average of 40,000 customers in the simulation, 86 replications have converged when Algorithm 2 is used, compared to 76 replications for the FH algorithm. For Algorithm 1, 73 replications have converged after an average of 69,200 customers, and for the GM-GP algorithm, 38 replications have converged after an average of 61,000 customers.)

Table 3 shows the same information when the smaller neighborhood structure N_2 given in Equation (20) is used. From this table we see that the performance of all of the algorithms is worse than that observed when using the larger neighborhood structure N_1. Again, Algorithm 2 gives the best performance in this setting. (For example, after an average of 200,000 customers in the simulation, 70 replications have converged when Algorithm 2 is used, compared to 29 replications for the FH algorithm. For Algorithm 1, 40 replications have converged after an average of 261,600 customers, and for the GM-GP algorithm, 22 replications have converged after an average of 222,000 customers.)

Tables 2 and 3 indicate that Algorithm 2 has the best performance when applied to solve the transient version of the optimization problem (18) using the parameter values given previously. The FH algorithm performs better than Algorithm 1 and the GM-GP algorithm if the larger neighborhood structure N_1 is used, but it performs worse than Algorithm 1 if the smaller neighborhood structure N_2 is used. Algorithm 1 always has better performance than the GM-GP algorithm. We conclude that using the larger neighborhood structure N_1 yields better performance for all four algorithms considered here than the smaller neighborhood structure N_2, because with the smaller neighborhood structure the underlying optimization problem has more local solutions.

Figure 5 **Performance of Algorithm 2 in the Transient Setting as a Function of the Parameter T Using the Neighborhood Structure N_2**

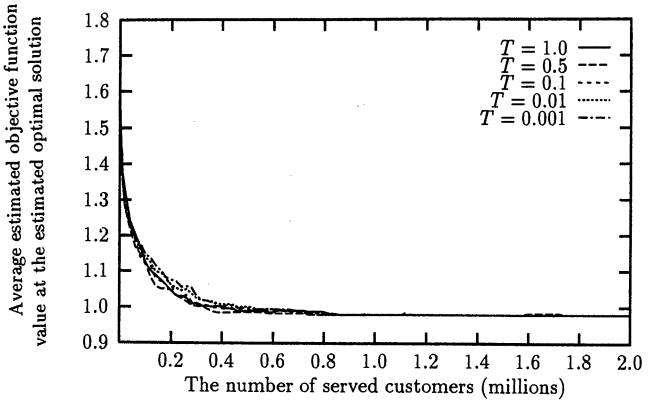


Table 2 **A Comparison of the Performance of Algorithms 1 and 2 and the GM-GP and FH Algorithms in the Transient Setting Using the Neighborhood Structure N_1**

	Algorithm 1		Algorithm 2		GM-GP Alg.		FH Alg.	
Iter. (k)	#	Cust.	#	Cust.	#	Cust.	#	Cust.
10	9	4,000	17	2,000	7	2,000	11	2,000
50	43	29,200	35	10,000	24	18,400	27	10,000
100	73	69,200	57	20,000	38	61,000	33	20,000
200	89	161,600	86	40,000	80	222,000	76	40,000
300	98	261,600	98	60,000	95	483,000	95	60,000
400	100	363,000	100	80,000	99	844,000	99	80,000
500	100	483,000	100	100,000	100	1,305,000	100	100,000
1,000	100	1,083,000	100	200,000	99	5,110,000	100	200,000

Table 3 **A Comparison of the Performance of Algorithms 1 and 2 and the GM-GP and FH Algorithms in the Transient Setting Using the Neighborhood Structure N_2**

	Algorithm 1		Algorithm 2		GM-GP Alg.		FH Alg.	
Iter. (k)	#	Cust.	#	Cust.	#	Cust.	#	Cust.
10	7	4,000	9	2,000	4	2,000	9	2,000
50	19	29,200	23	10,000	10	18,400	22	10,000
100	29	69,200	28	20,000	16	61,000	25	20,000
200	36	161,600	34	40,000	22	222,000	28	40,000
300	40	261,600	43	60,000	31	483,000	29	60,000
400	41	363,000	50	80,000	33	844,000	29	80,000
500	42	483,000	55	100,000	36	1,305,000	29	100,000
1,000	47	1,083,000	70	200,000	37	5,110,000	29	200,000

Tables 4 and 5 compare the performance of Algorithms 1 and 2 with that of the GM-GP and FH algorithms when they are applied to solve the steady-state version of the optimization problem (18) using the parameter values $T = 0.01$, and for all $k \in \mathbb{N}$, $T_k = 0.1/\ln(10 + k)$, $L_k = 50 + \lfloor 10 \ln(10 + k) \rfloor$ for Algorithms 1 and 2 and the FH algorithm, and $L_k = 50 + \lfloor k^2/200 \rfloor$ for the GM-GP algorithm (note that if it is necessary to estimate $f^{(s)}(x)$ in iteration k of the FH algorithm, we use Equation (17) to generate $W_{L_1}(x), \ldots, W_{L_k}(x)$ and average $\sum_{i=1}^{L_k} W_i(x)/L_k$ with the estimates of $f^{(s)}(x)$ generated in the previous iterations to obtain a new estimate of $f^{(s)}(x)$). Table 4 shows the results obtained by using the neighborhood structure N_1 described in Equation (19). Note that in the steady-state setting we use the same increasing sequence $\{L_k\}$ that satisfies Assumption 5 for both Algorithms 1 and 2, so column 4 of the table shows the number of customers used so far by each of these algorithms. It is clear from this table that Algorithm 2 has the best performance; the FH algorithm is the second best, followed by Algorithm 1; finally, the GM-GP algorithm needs a lot of computational effort before all 100 replications converge.

Table 5 shows the same information when the smaller neighborhood structure N_2 given in Equation (20) is used. Again, Algorithm 2 outperforms Algorithm 1, the FH algorithm performs less well than Algorithms 1 and 2, and the GM-GP algorithm yields the worst performance in this setting. Tables 2 through 5 indicate that the performance of Algorithm 2 is superior to the performance of the other three algorithms in this setting and that Algorithm 1 always shows better performance than the GM-GP algorithm, and has better performance than

Table 4 **A Comparison of the Performance of Algorithms 1 and 2 and the GM-GP and FH Algorithms in the Steady-state Setting Using the Neighborhood Structure N_1**

				GM-GP Alg.		FH Alg.	
Iter. (k)	Alg. 1 #	Alg. 2 #	Cust.	#	Cust.	#	Cust.
10	11	14	1,532	7	1,000	14	1,532
50	26	32	8,416	8	5,388	24	8,416
100	42	53	17,800	15	13,302	34	17,800
200	58	87	37,824	22	46,704	80	37,824
300	78	98	58,838	38	120,206	93	58,838
400	81	100	80,502	67	253,808	100	80,502
500	83	100	102,668	86	467,510	100	102,668
1,000	95	100	218,316	100	3,437,520	100	218,316
2,000	100	100	463,322			100	463,322
5,000	100	100	1,248,216			100	1,248,216

Table 5 **A Comparison of the Performance of Algorithms 1 and 2 and the GM-GP and FH Algorithms in the Steady-state Setting Using the Neighborhood Structure N_2**

				GM-GP Alg.		FH Alg.	
Iter (k)	Alg. 1 #	Alg. 2 #	Cust.	#	Cust.	#	Cust.
10	8	14	1,532	3	1,000	9	1,532
50	17	26	8,416	8	5,388	25	8,416
100	20	31	17,800	7	13,302	26	17,800
200	25	33	37,824	16	46,704	28	37,824
300	25	40	58,838	22	120,206	28	58,838
400	25	46	80,502	34	253,808	28	80,502
500	32	50	102,668	34	467,510	28	102,668
1,000	38	63	218,316	35	3,437,520	28	218,316
2,000	50	85	463,322			28	463,322
5,000	63	98	1,248,216			28	1,248,216

the FH algorithm when the smaller neighborhood structure N_2 is used. All the algorithms behave worse when the smaller neighborhood structure N_2 that yields many local solutions is used, particularly the GM-GP and FH algorithms. We believe that this difference in performance is primarily explained by the facts that Algorithms 1 and 2 use a constant (rather than decreasing) temperature and they use different approaches for estimating the optimal solution that do not require the Markov chain $\{X_k\}$ generated by the algorithms to converge to the set of global optimal solutions. Therefore, Algorithms 1 and 2 can escape from the local solutions more quickly than the GM-GP and FH algorithms.

6. Conclusion

In this paper, we presented a new method for solving discrete stochastic optimization problems. Our method resembles the original simulated annealing method that was designed for solving discrete deterministic optimization problems. We have shown that our method converges almost surely to the set of global optimal solutions under mild conditions. We have discussed how the convergence of our method can be accelerated by using a different approach for estimating the optimal solution. We also discussed how our methods can be used for solving discrete optimization problems when the objective function

values are estimated using either transient or steady-state simulation.

The performance of our methods depends on the choice of a number of parameters, including the temperature T, the neighborhood structure, and the number of estimated objective function values obtained in the different iterations. Our numerical results indicate that our simulated annealing methods do not appear to be very sensitive to the choice of the temperature T. In our particular problem, good choices of T are $0.001 \leq T \leq 0.1$ for Algorithm 1 and $0.001 \leq T \leq 1.0$ for Algorithm 2. We also note that in our example, the algorithms give better performance using the larger neighborhood structure, because the smaller neighborhood structure yields many local solutions. Moreover, our numerical results show that the use of a slowly increasing sequence $\{L_k\}$ appears to yield good performance of our algorithms, where L_k is the number of observations used to estimate the objective function values in iteration k.

Our numerical results also indicate that for our particular example and choice of parameter values, the simulated annealing algorithms proposed and analyzed in this paper appear to have better overall performance than the other two simulated annealing algorithms designed for solving discrete stochastic optimization problems (the method analyzed by Gelfand and Mitter and by Gutjahr and Pflug and the method analyzed by Fox and Heine). Finally, the modified simulated annealing algorithm that uses the state with the best average estimated objective function value as the estimate of the optimal solution performs particularly well in our numerical results. While the numerical results presented in this paper are representative of our overall experience with the four simulated annealing algorithms considered here, additional numerical experimentation is needed to determine whether the conclusions of our numerical study hold for other discrete stochastic optimization problems and other choices of the parameters of these algorithms. Moreover, research focused on determining and comparing the theoretical convergence rate of these algorithms would be valuable.[1]

[1] This research was supported in part by NSF under grants DDM-9210679 and DMI-9523111. The authors thank Pierre L'Ecuyer, the Associate Editor, and two anonymous referees for their helpful suggestions on this paper.

References

Alrefaei, M. H., S. Andradóttir. 1995. A new search algorithm for discrete stochastic optimization. C. Alexopoulos, K. Kang, W. R. Lilegdon, D. Goldsman, eds. *Proc. 1995 Winter Simulation Conf.*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 236–241.

——, ——. 1998. A modification of the stochastic ruler method for discrete stochastic optimization. Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology.

Aluffi-Pentini, F., V. Parisi, F. Zirilli. 1985. Global optimization and stochastic differential equations. *J. Optim. Theory and Appl.* **47** 1–16.

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

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

——. 1998. Accelerating the convergence of random search methods for discrete stochastic optimization. Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology.

Asmussen, S. 1987. *Applied Probability and Queues*. Wiley, Chichester, UK.

Bulgak, A. A., J. L. Sanders. 1988. Integrating a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems. M. Abrams, P. Haigh, J. Comfort, ed. *Proc. 1988 Winter Simulation Conf.*, Institute of Electrical and Electronics Engineers, Piscataway, NJ. 684–690.

Černý, V. 1985. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Theory and Appl.* **45** 41–51.

Devroye, L. P. 1976. On the convergence of statistical search. *IEEE Trans. Systems, Man, and Cybernetics* **SMC-6** 46–56.

Fox, B. L., G. W. Heine. 1995. Probabilistic search with overrides. *Ann. Appl. Prob.* **5** 1087–1094.

Gelfand, S. B., S. K. Mitter. 1989. Simulated annealing with noisy or imprecise energy measurements. *J. Optim. Theory Appl.* **62** 49–62.

Geman, S., D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intelligence* **PAMI-6** 721–741.

——, C.-R. Hwang. 1986. Diffusions for global optimization. *SIAM J. Control and Optim.* **24** 1031–1043.

Gurin, L. S. 1966. Random search in the presence of noise. *Engrg. Cybernetics* **4** 252–260.

Gutjahr, W. J., G. Ch. Pflug. 1996. Simulated annealing for noisy cost functions. *J. Global Optim.* **8** 1–13.

Haddock, J., J. Mittenthal. 1992. Simulation optimization using simulated annealing. *Comput. and Indust. Engrg.* **22** 387–395.

Hajek, B. 1988. Cooling schedules for optimal annealing. *Math. Oper. Res.* **13** 311–329.

——, G. Sasaki. 1989. Simulated annealing—to cool or not. *Systems & Control Letters* **12** 443–447.

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Kushner, H. J. 1963. Hill climbing methods for the optimization of multiparameter noise disturbed systems. *Trans. ASME, J. Basic Engrg.* **85** 157–164.

Marti, K. 1982. Minimizing noisy objective functions by random search methods. *Zeitschrift für Angewandte Mathematik und Mechanik* **62** T377–T380.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller. 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21** 1087–1092.

Mitra, D., F. Romeo, A. Sangiovanni-Vincentelli. 1986. Convergence and finite-time behavior of simulated annealing. *Adv. Appl. Prob.* **18** 747–771.

Shedler, G. S. 1993. *Regenerative Stochastic Simulation.* Academic Press, Boston, MA.

Wolff, R. W. 1989. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, Englewood Cliffs, New Jersey.

Yakowitz, S., E. Lugosi. 1990. Random search in the presence of noise, with application to machine learning. *SIAM J. Sci. Statist. Comput.* **11** 702–712.

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

Accepted by Pierre L'Ecuyer; received July 7, 1997. This paper has been with the authors 9 months for 2 revisions.