Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

SIMULATION OPTIMIZATION WITH HYBRID GOLDEN REGION SEARCH

Alireza Kabirian

College of Business and Public Policy

3211 Providence Drive
University of Alaska-Anchorage
Anchorage, AK 99508, USA

Sigurdur Olafsson

Dept. of Industrial and Manufacturing Systems Engineering
Black Engineering 3004
Iowa State University
Ames, IA 50011, USA

ABSTRACT

Simulation Optimization (SO) is a class of mathematical optimization techniques in which the objective function could only be numerically evaluated through simulation. In this paper, a new SO approach called Golden Region (GR) search is developed for continuous problems. GR divides the feasible region into a number of (sub) regions and selects one region in each iteration for further search based on the quality and distribution of simulated points in the feasible region and the result of scanning the response surface through a metamodel. The experiments show the GR method is efficient compared to three well-established approaches in the literature. We also prove the convergence in probability to global optimum for a large class of random search methods in general and GR in particular.

1 INTRODUCTION

Consider a chemical reaction in which a certain amount of materials are mixed together to produce a final product. The best final product would be as much acidic (measured by PH) as possible. A chemist may want to conduct a number of lab experiments with different combinations of materials in order to find the combination resulting in the highest PH. Assume the test is costly and destructive, that is the expensive consumed materials in an experiment could not be retrieved. From an operations research point of view, this is a numerical optimization problem with expensive and noisy objective function.

Generally, the concept of the chemical reaction experiment may be substituted with a fluids-mechanic simulation experiment in mechanical engineering, system dynamics simulation in business and economic modeling, war gaming in military, surveying experiment in statistics/sociology and the test of a new vaccine on animals in veterinary or medicine. In these experiments, one may be interested in minimizing experiments because they are monetarily expensive, time consuming or computationally intensive, practically dangerous, or ethically unjustifiable. Such cases may also arise when discrete-event simulation is applied to evaluate performance measure(s) of selected design points of a stochastic system. In these cases, the objective function may not be obtained in closed form. Hence, Simulation Optimization (SO) methods try to find the best designs of a system through numerical estimations of the performance measure(s) of the underlying system obtained via simulation. Our fundamental assumption in this paper is that the (simulation) experiment required to evaluate the objective function is very expensive.

Many approaches have been proposed in the literature for SO problems. Detailed review of the available methods can be found in Andradottir (1998a, 1998b, 2006), Olafsson and Kim (2002), Gosavi (2003), Fu et al. (2005) and Olafsson (2006). When the number of design options is limited, statistical selection methods are usually appropriate (Kim and Nelson 2006). Metamodel based methods such as Response Surface Methodology fit regression or neural networks on the response surface locally or globally and use it to conduct new searches (Barton and Meckesheimer 2006). A major class of SO approaches is random search methods such as stochastic ruler method (Yan and Mukai 1992), stochastic comparison algorithms (Gong et al. 1999) and COMPASS (Hong and Nelson 2006). The Model Reference Adaptive Search method (MRAS) is recently introduced in Hu et al. (2007). This method considers a probability distribution model for the location of the global optimum and tries to pile up the density of this distribution around the global optimum by periodically updating the parameters of the distribution. Gradient search methods such as Stochastic Approximation (Robins and Morono 1951 and Kiefer and Wolfowitz 1952) estimate the gradient of the objective function (Fu 2006) and then use gradient methods of mathematical programming. Metaheuristic methods such as Simulated Annealing (Krikpatrick et al. 1983), Genetic Algorithm (Holland 1992),

Tabu Search (Glover 1989, 1990) and Scatter Search (Glover 1997) are a rich set of deterministic optimization algorithms which are extensively used in practical SO problems and commercial packages (Olafsson 2006).

Among recent metaheuristics, the Nested Partitioning (NP) (Shi and Olafsson 2000a) has been found to be efficient for combinatorial optimization. This method partitions the solution space into a number of regions; then one of the regions is selected for further search and partitioning based on evaluating the quality of a number of sampled points from each region. Shi and Olafsson (2000b) extend the metaheuristic for stochastic optimization and Olafsson (2004) improves the efficiency of the method for SO problems by combining NP with Ranking and Selection (R&S). Kabirian and Olafsson (2007) and Kabirian (2006) propose Adaptive Partitioning Search (APS) method with similar partitioning idea for continuous problems with 2 decision variables. This time, instead of costly sampling from each region, APS selects promising partition based on the quality of already simulated points near each region (a local search strategy), the evaluation of a globally fitted metamodel (global search strategy) and the largeness of each region (exploration strategy). In this paper, we generalize APS to higher dimensions and combine it with appropriate R&S procedures to guarantee convergence.

The remainder of the paper is organized as follows. In section 2, we introduce a large class of continuous optimization methods called Probabilistic Search algorithms, link them with R&S and prove their convergence. We then propose our new iterative heuristic search based method called Golden Region in section 3 and 4. We prove in section 5 that the new method is a probabilistic search algorithm and conclude that it converges. Section 6 discusses practical efficiency of the method. Finally, concluding remarks and future search directions are presented in section 7.

2 ANALYSIS OF HYBRID PROBABILISTIC SEARCH METHODS

2.1 Probabilistic Search

Consider the deterministic continuous optimization problem $\min_{\theta\in\Theta} f(\theta)$ where $f:\Theta\to\mathbb{R}$ and $\Theta\subset\mathbb{R}^n$ is the compact set of feasible points. Let $\theta^*\in\Theta$ be one of the possibly many global optima of this problem.

Random Search methods are a class of iterative optimization algorithms in whose k-th iteration, a finite number of points $\theta_k^{(1)},\theta_k^{(2)},\dots,\theta_k^{(H_k)}\in\Theta$ are selected via a specific sampling strategy Ψ_k and evaluated (see Andradottir 2006 for SO version).

Algorithm 1 *Random Search Methods*

Step 0: (Initialize). Choose the initial sampling strategy Ψ_1 *and let* $k=1$.

Step 1: (Sample). Select points $\theta_k^{(1)},\theta_k^{(2)},\dots,\theta_k^{(H_k)}\in\Theta$ *according to sampling strategy* Ψ_k

Step 2: (Evaluate). Compute $f\left(\theta_k^{(j)}\right)$ *for* $j=1,2,\dots,H_k$.

Step 3: (Update). Considering the quality of the evaluated points thus far, pick and introduce current optimum of the search process. If termination condition(s) of the algorithm hold(s), stop the algorithm, otherwise choose an updated strategy Ψ_{k+1} *, let* $k=k+1$ *and go to step 1.*

This class is broad enough to include many heuristic and metaheuristic methods. We narrow the definition a bit to what we call Probabilistic Search (PS) methods as follows (Kabirian 2009a):

Definition 1 *Let* W *be a Lebesgue measure. The Probabilistic Search methods are a subclass of Random Search methods such that for any arbitrary subset* $G\subseteq\Theta$ *with* $W(G)>0$*, the followings hold:*

1) $Pr\left\{\bigcup_{j=1}^{H_k}\left(\theta_k^{(j)}\in G\right)\right\}>0 \quad \forall\, k=1,2,\dots$

2) $\lim_{k\to\infty} Pr\left\{\bigcup_{j=1}^{H_k}\left(\theta_k^{(j)}\in G\right)\right\}\neq 0.$

The problem of interest in this paper is minimization of a stochastic and expensive objective function defined below

$$\min_{\theta\in\Theta}\left\{f(\theta)=\mathrm{E}\left(L(\theta)\right)\right\} \tag{1}$$

where L is a function of decision variables and random variables of a stochastic system and it is an estimator of f, also $\mathrm{E}(.)$ is the mathematical expectation operator. Indeed, we assume the closed form of function $f(\theta)$ is not available and can only be numerically estimated by $L(\theta)$ through averaging a number of independent and identically distributed sample performance functions obtained via simulating design point θ.

2.2 Hybrid Probabilistic Search

The advantage of defining PS methods in Definition 1 is that if a PS method is merged with an Indifference Zone (IZ) procedure, the asymptotic convergence of the combined algorithm can be shown under a regular assumption explained below.

Assumption 1 *A ball* $\mathrm{B}(\theta^*;r)=\{y\in\mathbb{R}^n \,|\, |\theta^*-y|\le r\}$ *exists such that* $W(\mathrm{B}(\theta^*;r)\cap\Theta)>0$ *and* $f(\theta)$ *is continuous for all* $\theta\in\mathrm{B}(\theta^*;r)\cap\Theta$.

We propose using IZ methods periodically during PS methods. The combination of these two, which we call Hybrid Probabilistic Search is outlined below.

Algorithm 2 *Hybrid Probabilistic Search*

Step 0: Define two sequences called error rate denoted by $\{\alpha_h\}_{h=1}^{\infty}$ *and IZ (parameter) denoted by* $\{\gamma_h\}_{h=1}^{\infty}$ *where* $\lim_{h\to\infty}\alpha_h=\lim_{h\to\infty}\gamma_h=0$, $0<\alpha_{h+1}\le\alpha_h\le 1$ *and* $0<\gamma_{h+1}\le\gamma_h$. *Define the number of replications between the IZ implementations and denote it by* τ.

Step 1: Choose the initial sampling strategy Ψ_1 *and let algorithm iteration counter* $k=1$ *and IZ implementation counter* $h=1$.

Step 2: Denote the introduced optimum of the algorithm after iteration k *by* $\hat{\theta}_k^*$. *Set* $\hat{\theta}_1^*=\{\}$.

Step 3: Select new points $\theta_k^{(1)},\theta_k^{(2)},\dots,\theta_k^{(H_k)}\in\Theta$ *according to sampling strategy* Ψ_k *and let* $\mathrm{Z}_k=\left\{\theta_{k'}^{(j)}\ for\ k'=1,2,\dots,k\ and\ j=1,2,\dots,Hk'\right\}$.

Step 4: For $j=1,2,\dots,H_k$ *simulate* $\theta_k^{(j)}$ *with* η *replications.*

Step 5: Let $R_k(\theta_{k'}^{(j)})$ *be the number of simulation replications done for* $\theta_{k'}^{(j)}$ *by the end of iteration* k. *Set replication counter* $R_k(\theta_k^{(j)})=\eta$ *for* $j=1,2,\dots,H_k$. *Set* $R_k(\theta_{k'}^{(j)})=R_{k-1}(\theta_{k'}^{(j)})$ *for* $k'=1,\dots,k-1$ *and* $j=1,\dots,H_{k'}$.

Step 6: Denote by $L_r(\theta_k^{(j)})$, *the objective function of* $\theta_k^{(j)}$ *estimated in* r*-th simulation replication. Set sample mean* $L(\theta_k^{(j)})=\sum_{r=1}^{R_k(\theta_k^{(j)})}\frac{L_r(\theta_k^{(j)})}{R_k(\theta_k^{(j)})}$ *for* $j=1,2,\dots,H_k$.

Step 7: If $k\neq\tau h$, *then let* $\hat{\theta}_k^*\in\arg\min_{\theta\in\mathrm{Z}_k}L(\theta)$ *and go to step 8; otherwise do:*

7.1. Design an IZ method and apply it to Z_k *such that a difference of* γ_h *or more in the mean objective functions of the best point in* Z_k *and all other points in* Z_k *is detected with probability* $1-\alpha_h$ *or more. For* $k'=1,2,\dots,k$ *and* $j=1,2,\dots,H_{k'}$, *let* $\lambda_{hjk'}$ *denote the total number of new simulation replications consumed in the current IZ implementation for* $\theta_{k'}^{(j)}\in\mathrm{Z}_k$ *on top of* $R_k(\theta_{k'}^{(j)})$ *old simulation replications available for this point and let* $R_k(\theta_{k'}^{(j)})=R_k(\theta_{k'}^{(j)})+\lambda_{hjk'}$.

7.2. Set $L(\theta_{k'}^{(j)})=\sum_{r=1}^{R_k(\theta_{k'}^{(j)})}\frac{L_r(\theta_{k'}^{(j)})}{R_k(\theta_{k'}^{(j)})}$ *for all* $\theta_{k'}^{(j)}\in\mathrm{Z}_k$. *Then let* $\hat{\theta}_k^*\in\arg\min_{\theta\in\mathrm{Z}_k}L(\theta)$.

7.3. Set $h=h+1$ *and go to step 8.*

Step 8: If a termination condition holds, introduce $\hat{\theta}_k^*$ *as the optimum and exit the algorithm. Otherwise choose an updated strategy* Ψ_{k+1}, *let* $k=k+1$ *and go to step 3.*

Generally, any IZ procedure that uses old simulation replications along with new replications and guarantees selection of the best with a given probability when the true objective function of the best and the rest of the designs are distanced at least by an IZ parameter could be used in Hybrid PS algorithm. Boesel et al. (2003) proposes such procedures. Bayesian methods are other alternatives in which a posterior probability of correct selection is guaranteed (Chick and Inoue 2001a and 2001b,

1999, 2000). However, as far as we are aware, all statistical selection of the best procedures in the literature (including in Boesel et al. 2003 and Chick and Inoue 2001a and 2001b) assume that simulation outputs are normally distributed. Nelson et al. (2001) and Nelson and Goldsman (2001) study the robustness of normality assumption and conclude that probability of correct selection could approximately be retained with mild departures from normality. The main justification for normal assumption in many simulation studies is that interesting simulation outputs are usually averages of a large number of observations; hence, central limit theorem suggests normality holds asymptotically. In addition, we need to assume that the second moment of the objective function for all feasible values of decision variable is finite.

As we said earlier, any Hybrid PS method can asymptotically converge to a global optimum of problem (1) under Assumption 1 as Theorem 1 below states.

Theorem 1 *If a hybrid PS method is applied to SO problem (1) under Assumption 1, the sequence* $\{f(\hat{\theta}_k^*)\}_{k=1}^{\infty}$ *converges in probability to* $f(\theta^*)$*, that is* $\lim_{k\to\infty} Pr\{|f(\hat{\theta}_k^*) - f(\theta^*)| < \varepsilon\} = 1$ *for all* $\varepsilon > 0$.

Proof: *See Kabirian (2009a).*

2.3 Example Procedure

In this section, we showcase the effectiveness of the Hybrid PS methods. Of course, any PS method could potentially be used here; but here we are more interested to see how well the wedding between IZ methods and PS algorithms work. Therefore, we select the simplest possible PS method which is called Naïve Random Search (NRS). The sampling strategy of NRS picks one point ($H_k = 1$ for $k = 1,2,\ldots$) uniformly randomly from the compact feasible region. The method is called "Naïve" because it ignores the information of past searches in future sampling strategies. When applied to SO problems, NRS introduces the point with lowest estimated objective function as the current optimum.

Algorithm 3 *Naïve Random Search*

Step 1: *Let* $k = 1$.

Step 2: *Select one point from the feasible region uniformly randomly. Denote this point by* $\theta_k^{(1)}$ *and let* $Z_k = \{\theta_{k'}^{(1)} \text{ for } k' = 1,2,\ldots,k\}$.

Step 3: *Simulate* $\theta_k^{(1)}$ *with* η *replications and let* $L(\theta_k^{(1)}) = \sum_{r=1}^{\eta} \frac{L_r(\theta_k^{(1)})}{\eta}$ *be the estimated objective function.*

Step 4: *Let* $\hat{\theta}_k^* \in \arg\min_{\theta \in Z_k} L(\theta)$.

Step 5: *If termination condition(s) of the algorithm hold(s), introduce* $\hat{\theta}_k^*$ *as the optimum and stop the algorithm; otherwise let* $k = k + 1$ *and go to step 2.*

We are interested in comparing the performance of NRS with the so called Hybrid NRS defined below.

Definition 2 *Hybrid NRS is a kind of Hybrid PS method in which the sampling strategy of each iteration selects one point uniformly randomly from the feasible region.*

In a simple experiments, we use the IZ procedure of Boesel et al. (2003) as the IZ procedure required for the Hybrid NRS. Also, we set $\eta = 2$, $\tau = 100$, $\alpha_1 = 0.50$, $\gamma_1 = 0.10$ and for $h = 1,2,\ldots$ we use $\alpha_{h+1} = 0.9\alpha_h$ and $\gamma_{h+1} = 0.9\gamma_h$. To accelerate the experiments, we replace simulation with a noisy objective function. Specifically, we use a closed form 2-dimensional objective function with a unique global optimum. Whenever the simulation output is required for a point, we generate a zero-mean normal random variable with variance 10 and add it to the objective function value computed via the closed-form formula. Figure 1 shows the objective function of the problem (1) used in our experiment. The decision variables are both bounded between 0 and 10. The global optimum of the problem is (5,5) with the objective function 1. We terminated the optimization process of both algorithms when a budget of 1000 simulation replications was spent. Both methods were run 10000 times in order to get robust results. For each algorithm, we computed the average of the expected value of the objective function of the introduced optimum after each simulation replication. Figure 1 shows values of these averages as the optimization process progresses.

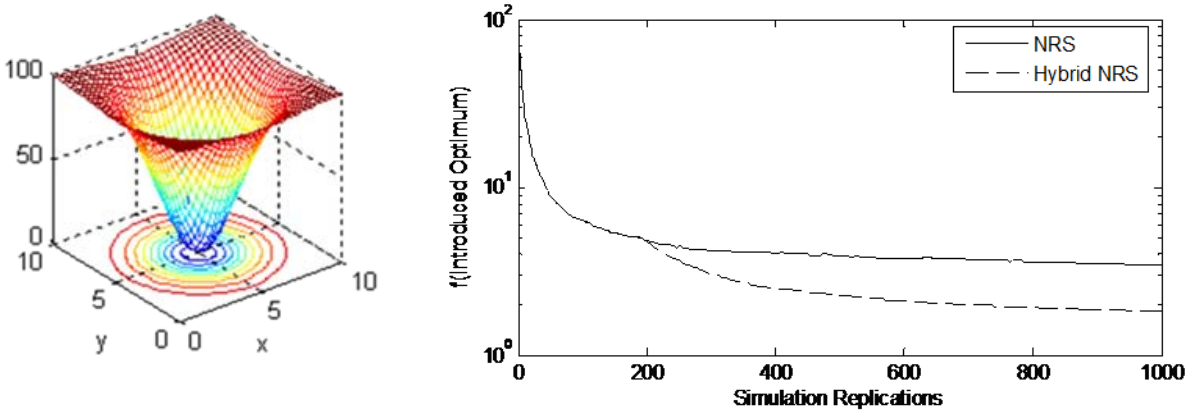

Figure 1: Left (The objective function of the test problem of HPS), Right (The performance curves of Naïve Random Search and Hybrid Random Search methods)

Figure 1 suggests that when IZ procedure is applied for the first time in Hybrid NRS, convergence is accelerated. As expected, the disadvantage of solely using NRS is that it may simulate a bad point which turn out to have a very good estimated objective function such that even near optimum points no longer could outperform the misleading estimated good quality of the bad point. In fact, IZ helps clean up the quality of the points NRS simulates.

In the next sections, we propose a new SO method called Golden Region search which is shown to be a PS method and hence benefits the convergence analysis studied in this section.

3 OVERVIEW OF THE GOLDEN REGION ALGORITHM

The Golden Region (GR) method is an iterative search-based optimization process (see Figure 2). Candidate solutions are simulated in a black box simulation module that returns estimations of objective functions to the optimizer. GR utilizes a meta-model learner that mimics the role of simulation. The benefit of training the metamodel is that the optimizer can utilize trained metamodel to cheaply evaluate the goodness of sampled points of feasible region before actually running expensive simulation.

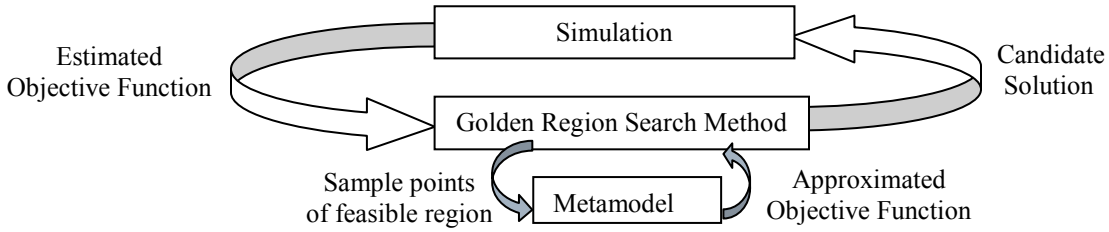

Figure 2: The framework of the method

The feasible region of the problem in the GR method is divided into a number of user defined sub regions which we simply call them regions in this paper. The algorithm starts out with a (random) population of feasible points. Simulating the design points of this initial population, the algorithm can proceed to an iterative search. In each iteration, one of the regions is selected, called promising region, then a new point in the promising region is chosen and simulated. Simulating this point, the iteration terminates and the algorithm continues until one of the terminating conditions holds. The key questions to be addressed here and in the next section are how the promising region and the new point inside it are selected.

In each iteration of the algorithm, a probability is assigned to each region, and then the promising region is selected randomly using these probabilities. we propose three criteria that affect the probability of selecting a particular region. These are called space score, meta-model score, and quality score. These three criteria are combined into a single indicator called total score for the region. The higher the total score, the more promising the region would be in terms of hiding the global optimum and a higher probability of selection for further search is assigned to the region.

The space score of a region represents how much the region has been visited. By assigning higher space score to the regions that are less visited, that is, have fewer number of already simulated points inside the region and other regions in close proximity of the region, the algorithm seeks to explore unvisited areas of the feasible region.

Motivated by the observation that running simulations is computationally expensive relative to other activities, GR method induces a global metamodel to the input-output of simulation. The metamodel score of a region is calculated using the metamodel based prediction of the objective functions of a number of sampled points inside the region. The better the predicted objective function of the samples of a region, the higher this criterion would be. In fact, this criterion monitors the quality of the whole feasible region.

Finally, the quality score for a region represents the quality of the estimated objective function of simulated points inside the region. The better the quality of these points, the higher the quality score. This criterion exploits the information of high quality simulated points with the hope of discovering local (or maybe global) optima near these points.

To construct a total score, we use a criterion coefficient for each of the three criteria, which are changed between iterations based on the effectiveness of using every criterion.

After selecting the promising region, a point is selected inside the promising region and simulated. Selection of this point is based on the most effective criteria of the promising region in the total score of the region, called dominating criterion, and consequently the probability of selecting the region. In fact, the algorithm needs to know why the promising region has been selected by determining the criterion which contributes most. If the dominating criterion is quality, it means the promising region has some high quality points that might be located in vicinity of a local optimum, so the new point is selected near the high quality point. If metamodel dominates, this is a signal that there are some promisingly good points in the promising region based on the global monitoring metamodel, so a promising point is simulated. Finally, in the case of space-criterion domination, the algorithm hopes to simulate a good quality random point in the unvisited area of the promising region. However, the algorithm is designed such that the new simulation point is not too close to already simulated points.

4 DETAILED GOLDEN REGION ALGORITHM

The feasible region of the problem Θ is partitioned into χ regions by the user in GR method. The space of i-th region is denoted by Θ_i. Kabirian and Olafsson (2008) provides some guidelines for partitioning the feasible region.

In the first step, a number of points, say υ points, are selected from the feasible region and simulated with a fixed number of replications. The initial population could be selected randomly or from certain regions of the feasible region to have a specific level of diversity in the population. After simulating initial population, the algorithm in each iteration selects one region called promising region among all regions, simulates a point inside the promising region and goes to next iteration if none of the terminating conditions hold (e.g. maximum possible iterations, maximum stalled searches, …). In the next sections, we discuss how the promising region and the point inside it are selected in a typical iteration k.

4.1 Promising Region Selection

As noted above, for selecting a new promising region, we use three criteria called space score, quality score and metamodel score. The algorithm assigns a value between zero and one for each criterion to each region based on some indicators and procedures that will be elaborated upon in the next sections. Denote by S_{ik}, Q_{ik} and M_{ik}, the values of space score, quality score and metamodel score assigned to i-th region respectively. We define a total score, denoted by ϕ_{ik} for region i using:

$$\phi_{ik} = \max\{s_k S_{ik}, q_k Q_{ik}, m_k M_{ik}\} \quad \forall i \tag{2}$$

where s_k, q_k and m_k are space, quality and metamodel criterion coefficients which take positive values and add up to one. In the first iteration, these coefficients are set to 1/3, but their values are changed at the beginning of the subsequent iterations based on the success or failure of the dominating criterion of the previous iteration in finding a good design point (see section 4.3 for details).

A probability of selection, denoted by π_{ik} for i-th region, is calculated for each region by normalizing the total score $\pi_{ik} = \frac{\phi_{ik}}{\sum_{i'=1}^{\chi} \phi_{i'k}} \quad \forall i$. Using a uniform random number between zero and one, a promising region for current iteration is then selected with respect to this probability mass function.

4.1.1 Space Score

In addition the feasible region partitioning, the algorithm requires a proximity measure for any two regions including the proximity of a region to itself in order to calculate the space score. This measure represents how similar the design points of one region are to those of the other region. We let $N_{ii'}$ denote the positive-valued proximity measure of each two regions where $i, i' = 1,2, \dots, \chi$. As an example of this measure, one may define a center point for each region (a vector of decision variables as the representative of the whole region), and relate the proximity measure of two regions to the reciprocal of the

Euclidian distance between the center points of the regions for $i \neq i'$ and choose the proximity function of each region with itself such that $\mathrm{N}_{ii} > \max_{i' \neq i} \mathrm{N}_{ii'}$.

We let T_{ik} denote the tally of the number of simulated points inside region i. We define an indicator, Y_{ik} called the visit indicator which measures the "density" of the simulated points inside and surrounding the region by $Y_{ik} = \frac{\sum_{i'=1}^{X} T_{i'k} \mathrm{N}_{ii'}}{\sum_{i'=1}^{X} \mathrm{N}_{ii'}} \quad \forall i$. Standardizing the visit indicator, we get the space score $S_{ik} = \frac{\max_{i'} Y_{i'k} - Y_{ik}}{\max_{i'} Y_{i'k} - \min_{i'} Y_{i'k}} \quad \forall i$.

4.1.2 Quality Score

Bayraksan and Morton (2006) provide procedures for constructing confidence intervals on optimality gap of a given good quality point. Here we use a simple procedure as follows to determine the quality of a region without spending more simulation efforts.

For $k \geq 0$, let $\mathrm{J}_k = \{1,2,\dots,v+k\}$ be the index set of simulated points up to iteration k. The standardized objective function of simulated point θ_j at the beginning of iteration k denoted by f'_{jk} is defined as $f'_{jk} = \frac{\bar{L}_k(\theta_j) - \min_{j'} \bar{L}_k(\theta_{j'})}{\max_{j'} \bar{L}_k(\theta_{j'}) - \min_{j'} \bar{L}_k(\theta_{j'})} \quad \forall j \in \mathrm{J}_{k-1}$ where $\bar{L}_k(\theta_j) = \frac{\sum_{r=1}^{R_k(\theta_j)} L_r(\theta_j)}{R_k(\theta_j)}$ is the sample mean. The standard objective function values are updated in iterations that best or worst found points change or estimates of already simulation points are renewed.

We need the following definition for this score.

Definition 3 *The saturation neighborhood of a point $y = (y_1, y_2, \dots, y_n) \in \mathbb{R}^n$ denoted by $\breve{E}_{\sigma,\xi}(y)$ is the region inside an ellipsoid centered at θ with scale parameter vector $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n) \in \mathbb{R}^n$ and radius $\xi \in \mathbb{R} \geq 0$, that is $\breve{E}_{\sigma,\xi}(y) = \left\{ x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \;\middle|\; \sum_{l=1}^{n} \left(\frac{x_l - y_l}{\sigma_l}\right)^2 \leq \xi^2 \right\}$.*

If there is at least one simulated point in region i, the Quality Score is:

$$Q_{ik} = \max_{\{\theta_j \in \Theta_i | j \in \mathrm{J}_{k-1}\}} U_{jk}\,(1 - f'_{jk}) \quad \forall i \tag{3}$$

otherwise it is set to zero.

In the above formula, U_{jk} that we call the saturation indicator of simulated point θ_j plays an important role. The key idea of the saturation indicator is that searches near already simulated points should be limited. If such searches were always allowed, the algorithm might excessively search surrounding local optima. The better the quality of the objective function of a point, the higher the maximum possible searches near this point, and consequently more searches are allowed in associated area. We call already simulated point θ_j "saturated" (and set $U_{jk} = 0$) if the number of simulated designs in the saturation neighborhood $\breve{\mathrm{E}}_{\sigma,\xi}(\theta_j)$ exceeds its maximum possible searches, otherwise the point is "unsaturated" (and $U_{jk} = 1$) and is allowed to contribute to the value of the quality score for the encompassing region (The parameters σ and ξ are defined by user). Mathematically, if u_{jk} represents the number of already simulated points in the saturation neighborhood of θ_j, the saturation indicator is then assigned by:

$$U_{jk} = \begin{cases} 1 & u_{jk} \leq U_{jk}^* \\ 0 & \text{o.w.} \end{cases} \quad \forall j \in \mathrm{J}_{k-1}$$

where U_j^* is the upper bound of allowed searches in the saturation neighborhood of j-th simulated point. For instance, a linear interpolation could be used to find U_{jk}^* for θ_j, that is $U_{jk}^* = 1 + (U^* - 1)\frac{\left(\max_{j'} \bar{L}_k(\theta_{j'}) - \bar{L}_k(\theta_j)\right)}{\max_{j'} \bar{L}_k(\theta_{j'}) - \min_{j'} \bar{L}_k(\theta_{j'})}$. Here, the number of possible searches for the best and worst found points are respectively set to a user defined parameter U^* and 1.

4.1.3 Metamodel Score

While the quality criterion acts locally, the metamodel criterion models the response surface globally and monitors the whole feasible region.

Before the first iteration, a population of (random) sample points should be selected from the space of each region. Whenever the metamodel is updated, such as the first iteration when the metamodel is initially constructed, it predicts the objective function of sampled points from each region. Let β_i be the number of metamodel samples in i-th region which is a us-

er-defined parameter. Let A_{ijk} be the objective function of the j-th sample of i-th region approximated by metamodel. Let $A'_{ijk} = \frac{A_{ijk} - \rho_L}{\rho_U - \rho_L}$ be the standardized value of A_{ijk} where ρ_L and ρ_U are user-defined parameters. The standardization makes sure A'_{ijk} values are positive and bounded, say $0 < \rho_L \leq A'_{ijk} \leq \rho_U < \infty$. The metamodel score is defined via $M_{ik} = \frac{\min_{i'}\{\min_j A'_{i'jk}\}}{\min_j A'_{ijk}} \quad \forall i.$

Many different global metamodeling techniques have been developed to mimic how simulation model transforms inputs into outputs. Traditional examples are regression and neural network, but other options include Bayesian metamodels and Kriging. For an example of suitable neural network metamodel, see Kabirian (2009b) and Kabirian and Olafsson (2008).

4.2 New Point Generation

After selecting a new promising region, a point inside this region must be simulated. Recall from (2) that the total score equals the largest product of each of the three scores and its coefficient. The largest value defines the dominating criterion and determines the generation of the next point.

If the dominating criterion is space, first β' sample points are selected randomly from the space of the promising region. Then, the sample point whose minimum distance with other already simulated points is maximum, is selected and simulated. This strategy tries to improve the significance of difference between the new simulated design point and other simulated designs.

If the dominating criterion is metamodel, the algorithm chooses the point inside the promising region that has had the best approximated objective function based on metamodel. Mathematically, if the code of the promising region is i , the sample point $j \in \arg\min_{j'} A'_{ij'k}$ of the promising region is simulated. Once the new point is simulated, a new sample point is drawn from the space of the promising region and is replaced with the new simulated point in the population of sample points of the promising region for metamodel.

Finally, if the dominating criterion is quality, then the best unsaturated simulated point of the promising region must first be found (see equation (3)). Denote this point by $\dot{\theta}_k \in \arg\max_{\theta_j \in \Theta_i} U_{jk}(1 - f'_{jk})$. Then a point is selected randomly with a multivariate normal distribution with mean $\dot{\theta}_k$ and covariance matrix ω which is a user defined matrix. If this random point is feasible, it is simulated, otherwise the process of generating a point with the same distribution and checking the feasibility is repeated until a feasible point is found and simulated.

4.3 Adaptive Criteria Coefficients

The initial values of the coefficients s_k, m_k and q_k in (2) are equal to 1/3, that is $s_1 = m_1 = q_1 = 1/3$, but these values may change at the beginning of next iterations based on the performance of the dominating criterion in previous iteration. The main rule here is that when quality (metamodel) dominates in an iteration and has served to find a "good" point, then the coefficient of the quality (metamodel) criterion is increased and those of the other two criteria are decreased. When a bad point is simulated with quality (metamodel) domination, the coefficient of the quality (metamodel) criterion is decreased and those of the other two criteria are increased. Whenever space dominates, we always decrease its coefficient and add to those of metamodel and quality criterion.

For $k = 1,2, \ldots$, we set $s_k + m_k + q_k = 1$ and $0 < \delta_L \leq s_k, m_k, q_k \leq \delta_U \leq 1$ where δ_L and δ_U are user defined parameters. Also, we update coefficients this way: $q_{k+1} = q_k + \Delta q_{k+1}$, $m_{k+1} = m_k + \Delta m_{k+1}$, $s_{k+1} = s_k + \Delta s_{k+1}$ where Δq_{k+1}, Δm_{k+1}, and Δs_{k+1} are the amount of change in associated coefficients defined as follows. We may face one of 3 cases (assume region i has been the promising region in iteration k):

Case 1. Quality is the dominating criterion of previous iteration, that is $\phi_{ik} = q_k Q_{ik}$. Assume j-th simulated point located in the promising region has been the best unsaturated point in iteration k and the new simulation point j' has been selected with respect to j.

a. If estimated objective function of point j' is better than point j'', that is $f'_{j'k} \leq f'_{jk}$ then the algorithm increases (incentivizes) the coefficient of Quality criterion and simultaneously decreases the coefficients of the other two criteria as follows:

$$\Delta q_{k+1} = \delta(\delta_U - q_k) \max\left\{\frac{f'_{jk} - f'_{j'k}}{f'_{jk}}, 1\right\}$$

$$\Delta m_{k+1} = -\frac{m_k - \delta_L}{s_k + m_k - 2\delta_L}\Delta q_{k+1}$$

$$\Delta s_{k+1} = -\frac{s_k - \delta_L}{s_k + m_k - 2\delta_L}\Delta q_{k+1}$$

where the parameter δ is a constant defined by user.

b. On the other hand, if the estimated objective function of new point j' is worse than the best unsaturated point j, that is $f_{j'k}^{'} > f_{jk}^{'}$ then the reverse way is done in which Quality Score is punished as follows:

$$\Delta q_{k+1} = -\delta(q_k - \delta_L)\max\left\{\frac{f_{j'k}^{'} - f_{jk}^{'}}{f_{jk}^{'}}, 1\right\}$$

$$\Delta m_{k+1} = \frac{\delta_U - m_k}{2\delta_U - s_k - m_k}\Delta q_{k+1}$$

$$\Delta s_{k+1} = \frac{\delta_U - s_k}{2\delta_U - s_k - m_k}\Delta q_{k+1}$$

Case 2. Metamodel is the dominating criterion of previous iteration. The similar idea as case 1 is used, but here the last simulated point is compared with the best found simulated point.

Case 3. Space is the dominating criterion of previous iteration. We always reduce the coefficient of space criterion as follows:

$$\Delta s_{k+1} = -\delta'(s_k - \delta_L)$$

$$\Delta m_{k+1} = \frac{\delta_U - m_k}{2\delta_U - q_k - m_k}\Delta s_{k+1}$$

$$\Delta q_{k+1} = \frac{\delta_U - q_k}{2\delta_U - q_k - m_k}\Delta s_{k+1}$$

where constant δ' is defined by user.

4.4 Stopping Conditions

Many stopping criteria could be defined for GR. The followings are just a handful of examples:

- When the number of iterations exceed a pre-specified level. Equivalently, when the number of simulation replications reaches a maximum.
- When the number of stalled searches reaches a threshold, i.e. when the number of subsequent introduced optima without any improvement in terms of estimated objective function reaches a maximum.
- When maximum total (simulation) optimization run time reaches.

An important question when the algorithm stops is which point should be introduced as the best. GR simply introduces the point with the best estimated objective function as the optimum, $\hat{\theta}_k^* \in \arg\min_{j \in \mathrm{J}_k} \bar{L}_k(\theta_j)$. However, when GR is merged with IZ procedures in the next section to form Hybrid GR, the introduced optima sequence will converge to the global optimum.

5 CONVERGENCE OF GOLDEN REGION SEARCH

It can be shown that GR satisfies the conditions of a PS method defined in Definition 1.

Theorem 2 *GR is a PS method.*
Proof: *See Kabirian (2009b) or Kabirian and Olafsson (2008).*

When GR is merged with IZ methods, the convergence is guaranteed.

Definition 4 *When GR is merged with IZ as the Hybrid PS Algorithm 2 outlines, the combination is called a Hybrid GR.*
Corollary 1 *By Theorems 1 and 2 and under Assumption 1 for problem (1), the objective function of introduced optima of any Hybrid GR converges in probability to that of the global optimum.*

6 NUMERICAL RESULTS

We have tested GR on a variety of objective functions and compared its performance with many other methods in the literature. Kabirian and Olafsson (2008) provide a statistical test design for comparing the efficiency of two or more SO methods. We have used this design to compare the efficiency of GR with Genetic Algorithm (GA), globalized Stochastic Approximation (SA) (see Kabirian 2009b or Kabirian and Olafsson 2008 for the algorithm) and Model Reference Adaptive Search method 1 (MRAS1) (Hu et al. 2005 and 2007) under two different settings:

1. With specialized parameters of each competing method on each test bed
2. With an overall (generic) parameter set for each method on all test beds.

Figure 3 shows one of the test problems we used in our experiments along with performance curves of 4 competing methods under above two settings. We do not use IZ add-on in GR method in these results. We only test the ability of the methods in hitting good quality points, not necessarily their ability to recognize the quality of the hit points. Therefore the y-axe shows the expected value of the best hit points.

Obviously, performance of all methods under specialized parameters are better than that under generic parameters. These graphs along with other computational results documented in Kabirian (2009b) and Kabirian and Olafsson (2008) suggest that GR is efficient when compared to other methods.

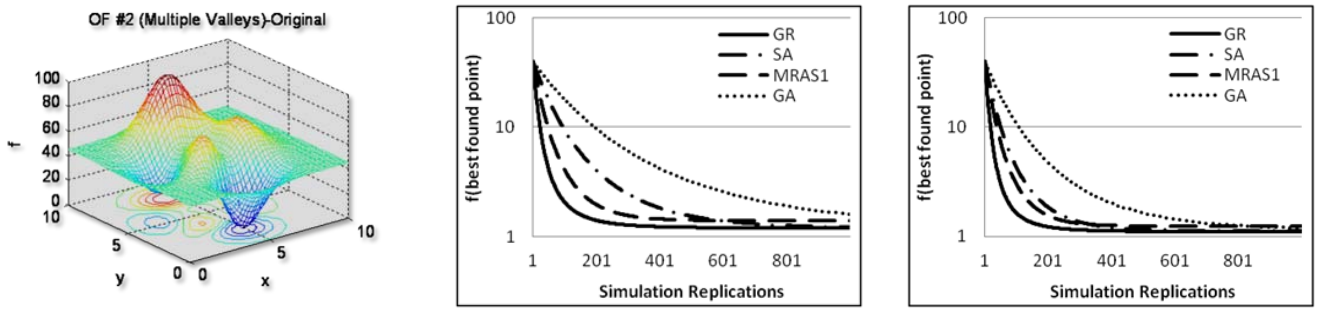

Figure 3: Left (Test Function), Middle(Results for Generic Parameters), Right (Results for Specialized Parameters)

7 REMARKS

In this paper, a new simulation optimization method called Golden Region search was developed based on some ideas to improve the efficiency of simulation optimization methods. GR is primarily aimed at continuous optimization problems. The method combines different search strategies to explore the feasible region. The proposed method requires partitioning of the feasible region and uses a metamodel along with the information of the distribution and goodness of past simulation points to find the GR containing the global optimum. We proposed a neural network framework as metamodel with structuring and training procedures. Merging the core optimizer of GR with R&S procedure, we prove the asymptotic convergence of the method. We tested the efficiency of merging R&S procedures with Random Search methods and the performance of GR.

GR has a generic framework which is very flexible for professional users of SO methods. However, we think it is critical to provide some guidelines for ordinary users to implement the method in practice. In Kabirian (2009b) and Kabirian and Olafsson (2008), we discuss three major implementation issues in details. These include guidelines for partitioning an arbitrary feasible region, tuning the parameters of the method and a neural network metamodel structuring and training procedure. Moreover, Kabirian (2009b) and Kabirian and Olafsson (2008) document more computational results which show GR works efficiently on different test problems with a set of educated guesses for its parameters. Therefore, we believe the method does not suffer too much from large set of its parameters.

A number of research directions remain open. The GR method can be extended to the problems with stochastic constraints (Kabirian and Olafsson 2009a and 2009b). Metamodeling can improve the efficiency of the SO methods in many ways (Barton and Meckesheimer 2006). We used a neural network approach proposed in Kabirian (2006) and Kabirian snd Olafsson 2008) with a specific updating procedure. The appropriateness of other metamodel updating procedures as well as metamodel types deserves more investigation. We used batch training procedure, however incremental training might be more efficient for our method.

Optimal Computing Budget Allocation (Chen and Lin 2000) is one of the potential ranking and selection methods that could be merged with GR to form a Hybrid GR. The idea is to define an increasing simulation budget periodically to spend on "cleaning up" the already simulated points aiming at increasing the approximated probability of correct selection.

We believe that extreme values of decision variables are more likely to be the optimal solutions of real SO problems. It might be beneficial if SO methods first try the extreme values before their main optimization routines. Since many of the heuristic methods of SO including our proposed method in this paper document use an initial population of points, the extreme solutions might be included in these populations.

REFERENCES

Andradottir, S. 2006. An overview of simulation optimization via random search. In *Handbooks in Operations Research and Management Science*, ed. S. G. Henderson and B. L. Nelson,13:617–632, Elsevier.

Andradottir, S. 1998a. A review of simulation optimization techniques. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan, 151–158. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Andradottir, S. 1998b. Simulation optimization. In *Handbook of Simulation,* ed. J. Banks, Chapter 9. Malden, Massachusetts, John Wiley and Sons Inc.

Barton, R. R., and M. Meckesheimer. 2006. Metamodel-based simulation optimization, In *Handbooks in Operations Research and Management Science*, ed. S. G. Henderson and B. L. Nelson, 13:617–632, Elsevier.

Boesel, J., B. L. Nelson, and S. Kim. 2003. Using ranking and selection to "clean up" after simulation optimization. *Operations Research* 51(5):814–825.

Chen, H. C., and J. Lin. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Journal of DEDS* 10(3):251–270.

Chick, S. E., and K. Inoue. 2001a. New two-stage and sequential procedures for selecting the best simulated system, *Operations Research* 49(5):732–743.

Chick, S. E., and K. Inoue. 2001b. New procedures for identifying the best simulated system using common random numbers, *Management Science* 47(8):1133–1149.

Fu, M. C. 2006. Gradient estimation. In *Handbooks in Operations Research and Management Science*, ed. S. G. Henderson and B. L. Nelson, 13:575–616, Elsevier.

Fu, M. C., F. W. Glover, and J. April. 2005. Simulation optimization: a review, new developments, and applications. In *Proceedings of the 2005 Winter Simulation Conference*. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 83–95. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Glover, F. 1997. A template for scatter search and path relinking. In *Lecture Notes in Computer Science,* ed. Hao J. K. , E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, 1363:1–53, Springer, Berlin.

Glover, F. 1989. Tabu search – part I. *ORSA Journal on Computing* 1:190–206.

Glover, F. 1990. Tabu search –part II. *ORSA Journal on Computing* 2:4–32.

Gong, W. B., Y. C. Ho, and W. Zhai. 1999. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization.* 10(2):384–404.

Gosavi, A. 2003. *Simulation-based optimization: parametric optimization techniques and reinforcement learning*. Kluwer Academic Publisher, Norwell, Massachusetts.

Holland, J. H. 1992. Genetic algorithms. *Scientific American* 267(1):66–72.

Hong, L. J., and B. L. Nelson. 2006. Discrete optimization via simulation using COMPASS. *Operations Research* 54(1):115–129.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. A model reference adaptive search algorithm for global optimization. *Operations Research* 55(3):549–568.

Hu, J., M. C. Fu, and S. I. Marcus. 2005. Stochastic optimization using model reference adaptive search. In *Proceedings of the 2005 Winter Simulation Conference*. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 811–818. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kabirian, A. 2006. *A review of current methodologies and developing a new heuristic method of simulation optimization*. Master of Science Dissertation, Sharif University of Technology, Tehran, Iran.

Kabirian, A. 2009a. Hybrid probabilistic search methods for simulation optimization, *Journal of Industrial and Systems Engineering*, 2(4):259–270.

Kabirian, A. 2009b. *Continuous optimization via simulation using Golden Region search*, PhD Dissertation, Iowa State University, Ames, Iowa.

Kabirian, A., and S. Olafsson. 2007. Allocation of Simulation runs for Simulation optimization, In *Proceedings of the 2007 Winter Simulation Conference,* ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 363–371.

Kabirian, A., and S. Olafsson. 2008. Simulation optimization with Golden Region search, Working paper, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa

Kabirian, A., and S. Olafsson. 2009a. Multiresponse Golden Region search under noisy constraints, Working paper, Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa

Kabirian, A., and S. Olafsson. 2009b. Selection of the best with stochastic constraints, To Appear In *Proceedings of 2009 Winter Simulation Conference,* ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls.

Kiefer, J., and J. Wolfoxitz. 1952. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics* 23(3):462–466.

Kim, S., and B. L. Nelson. 2006. Selecting the best system. In *Handbooks in Operations Research and Management Science*, ed. S. G. Henderson and B. L. Nelson, 13:501–534, Elsevier.

Krikpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220(4598):671–680.

Nelson, B. L., and L. D. Goldsman. 2001. Comparison with a standard in simulation experiments. *Management Science* 47(3):449–463.

Nelson, B. L., J. Swann, D. Goldsman, and W. Song. 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research* 49(6):950–963.

Olafsson, S., and J. Kim. 2002. Simulation optimization. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1931–1936. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Olafsson, S. 2004. Two stage nested regions method for stochastic optimization. *Methodology and Computing in Applied probability* 6(1):5–27.

Olafsson, S. 2006. Metaheuristics. In *Handbooks in operations research and management science*, ed. S. G. Henderson and B. L. Nelson, 13:633–654, Elsevier.

Robins, H., and S. Monro. 1951. A Stochastic approximation method. *Annals of Mathematical Statistics* 22(3):400–407.

Shi, L., and S. Olafsson. 2000a. Nested partitions method for global optimization. *Operations Research* 48(3):390–407.

Shi, L., and S. Olafsson. 2000b. Nested partitions method for stochastic optimization. *Methodology and Computing in Applied probability* 2(3):271–291.

Yan, D., and H. Mukai. 1992. Stochastic discrete optimization. *SIAM Journal on Control and Optimization* 30(3):594–612.

AUTHOR BIOGRAPHIES

ALIREZA KABIRIAN is an assistant professor in the College of Business and Public Policy at the University of Alaska-Anchorage. In 2009, he received his Ph.D. in Industrial Engineering from Iowa State University. He also obtained his degrees of M.Sc. and B.Sc. in Industrial Engineering respectively from Sharif University of Technology and Azad University of Najafabad in Iran. His specific research interests are simulation optimization and energy planning. He is a member of INFORMS and IIE. His e-mail address is <a_kabirian@yahoo.com>.

SIGURDUR OLAFSSON is an associate professor in the Department of Industrial and Manufacturing Systems Engineering at Iowa State University. He received a M.S. and Ph.D. from the Department of Industrial Engineering at the University of Wisconsin – Madison, and B.S. in mathematics from the University of Iceland. His research interests focus on discrete optimization, especially metaheuristics, and their uses in simulation optimization and data mining. His web page can be found at <www.public.iastate.edu/~olafsson>.