Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

COMBINING STRONG AND SCREENING DESIGNS FOR LARGE-SCALE SIMULATION OPTIMIZATION

Kuo-Hao Chang
Ming-Kai Li

National Tsing Hua University
30013 Hsinchu Taiwan

Hong Wan

Purdue University
West Lafayette, IN 47907-2023, USA

ABSTRACT

Simulation optimization has received a great deal of attention over the decades, which probably can be attributed to its generality and solvability in many practical problems. On the other hand, simulation optimization is well-recognized as a difficult problem, especially when the problem dimensionality grows. STRONG is a newly-developed method built upon the traditional response surface methodology. Its advantages lie in the automation and provable convergence, as opposed to traditional RSM that requires human involvements and the final solution has no quality guarantee. Moreover, the use of efficient experimental design and regression analysis grants STRONG the great potential to deal with large-scale problems. This paper exploits the basic structure of STRONG and integrates an efficient screening design to handle problems that are of realistic scale, i.e., hundreds of factors. The convergence of the new algorithm is proved. The computational advantage is shown by numerical evaluations.

1 INTRODUCTION

Simulation optimization is concerned with identifying optimal design parameters for a stochastic system, where the objective function is expressed as an expectation of a function of response variables associated with a simulation model. Mathematically, the optimization problem can be written as:

$$\text{minimize } g(x), \; x \in \mathbb{X}, \tag{1}$$

where $g(x) = E(G(x,\omega))$ is the objective function, e.g., the expected waiting time; $G(x,\omega)$ is the stochastic output generated by the simulation model, e.g., the averaged waiting time; ω refers to the stochastic effect of the system, x is a controllable factor (possibly a vector); and $\mathbb{X}$ is the parameter space, either defined explicitly or implicitly. Although many real-world problems can be cast in the framework of simulation optimization, three common features in these problems create significant challenges for acquiring quality solutions: (1) large number of factors—many practical simulation models include many factors (hundreds and thousands); (2) complex factor interactions— the interacting relationship among factors is nonlinear, complex and unknown; the only way to estimate it is through Input/Output (I/O) of the simulation model, and (3) randomness—the simulation output is stochastic in nature. The objective function can only be estimated with random errors.

In this paper, we attempt to develop a framework that can address these three issues. In particular, we are focused on unconstrained simulation optimization with continuous decision variables. We assume the simulation model is a black box; namely there is no way to obtain a close-form expression for the objective function, or derive an effective gradient estimator for purposes of optimization. Contemporary methodologies for handling this class of problem include stochastic approximation, response surface methodology and the many metaheuristics (such as genetic algorithms, tabu search and simulated annealing) (Fu 2002). Stochastic approximation (Robbins and Monro 1951) (SA) is one the most well-known algorithm for simulation optimization. It is a stochastic counterpart of the renowned steepest descent

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

method in (deterministic) nonlinear programming. Depending on how the gradient is estimated in the algorithm, SA has several versions. For example, FDSA (finite-difference stochastic approximation) and SPSA (simultaneous perturbation stochastic approximation) correspond to the SA algorithms where finite-difference method and simultaneous perturbation method are used to estimate the gradient respectively (Spall 2003). The advantages of SA is that the algorithm is easy to understand and implement; moreover, it does not require substantial knowledge of the simulation model under study. Under certain regularity conditions, SA can also be proved to converge to one of the (local) optimal solution of the original problem (Kushner and Yin 2003)(Spall 2003). However, SA also has several disadvantages, thereby limiting its applications. First, the algorithm performance is very sensitive to the gain sequence (the iterative step size of the algorithm), which is determined by users. Selecting a good gain sequence is never an easy task. Second, SA usually works well for unimodal-structured response surfaces. When the response surface is multimodel, however, the performance is not satisfactory.

Response surface methodology (Myers et al. 2009) (RSM) is another popular choice for simulation optimization. RSM uses a sequential experimentation strategy to build a series of local polynomial regression model to approximate the underlying response surface. RSM applies many powerful statistical tools, such as design of experiments and regression analysis, in its framework and therefore enjoys considerable computational advantages, especially for large-scale problems. However, traditional RSM is a heuristic algorithm in which the quality of final solution cannot be quantified. Also, RSM requires experimenters' opinions in the search process, which is impractical when the number of computer iteration is large.

In Chang et al. (2007), we proposed a new RSM-based algorithm called STRONG (Stochastic Trust-Region Response Surface Method) that borrows the concept of "trust region" used in Trust Region Method (Conn, Gould, and Toint 2000) developed in nonlinear programming to replace the "region of interest" in traditional RSM. We showed that STRONG is an automated algorithm and proved that STRONG has a nice convergence property. In Chang and Wan (2009), we further extended the STRONG algorithm, called STRONG-X, to handle the response surface that is generally-distributed. STRONG is attractive because it eliminates the disadvantages, yet maintains the advantages, of traditional RSM. However, currently STRONG employs central composite design (CCD) to fit a second-order model, which can be computationally demanding when solving practical large-scale problems. For example, suppose we want to find the optimal design for a stochastic system with 100 factors and each simulation takes one minute to run. Building a second-order model based on a CCD requires at least 3 days, which is clearly not feasible in practice.

In this research, we improve the current STRONG framework so that it can efficiently handle large-scale problems frequently encountered in practice. The basic idea is to combine efficient screening designs with STRONG to eliminate unimportant factors so that the subsequent local model development and optimization can focus on the most influential ones, resulting in significant computation reductions. In the literature, many screening strategies have been proposed to identify important factors, for examples, Trocine and Malone (2000), Trocine and Malone (2001) and Morris (2006). Different from traditional physical experiment-based screening designs, simulation experiments have unique properties; therefore require special screening designs. *Controlled Sequential Bifurcation* (CSB)(Wan et al. 2006), is a new group screening method specifically designed for stochastic simulation experiments. CSB takes full advantage of sequential nature of simulation experiments and screens factors in groups: If the group effect is unimportant, all factors in the group are considered to be unimportant; if the group effect is important, the factors in the group will be split into two smaller subgroups for further testing. The appeal of CSB lies in that it can control the Type I error for each factor, i.e, the probability of an unimportant factor being classified as important, as well as the power at each bifurcation step, i.e., the probability of a critical group being identified. Numerical evaluations show that CSB requires much less simulation runs compared to traditional screening designs, especially when the number of factors is large and the fraction of important factors is small. A series of further studies have been conducted and several versions of related factor screening designs are developed, such as CSB-X (Wan et al. 2010), CSFD (Shen and Wan 2009), and FF-CSB (Sanchez et al. 2009). Among these approaches, CSB-X and CSFD are found to have a complementary relationship, i.e., the advantage of CSB-X is the disadvantage of CSFD and vice versa. Shen et al. (2010) therefore proposed a hybrid approach that combines CSB-X and CSFD to enjoy the benefits from both sides.

This paper integrates STRONG with efficient screening designs to handle large-scale problems. Specifically, for each iteration, the developed framework, STRONG-LS, applies the hybrid screening method to identify the important factors, followed by the local model development based on the selected factors and solving for the solution that optimizes the local model. If the solution is satisfactory, STRONG-LS moves to it and starts over the whole procedure again. Otherwise, STRONG-LS will improve the quality of local model by including more factors, selecting a more elaborate local model for fitting the response surface (e.g., the second-order model), or increasing the number of design points, until a satisfactory solution is found. Because the large-scale problem is broken into several small ones, the difficulty for solving the problem is significantly reduced. Moreover, for problems where *sparsity of effects principle* holds (Montgomery 2009), the screening design employed in STRONG-LS results in significant computational gains compared to other competing algorithms. The numerical evaluation will be shown in Section 3.

It is worth mentioning that the screening design employed in STRONG-LS differs from traditional RSM in the following two aspects. First, the first stage of traditional RSM is usually factor screening, which is typically based on a first-order design, such as Plackett-Burman design (Myers et al. 2009), while STRONG-LS employs the hybrid screening method that takes advantage of the highly sequential nature of simulation experiments. Second, the factor screening in traditional RSM is only done once in the whole process. In STRONG-LS, however, the factor screening method is performed iteratively because the influential factors are likely to vary as the algorithm moves to different locations. We take a simple queueing example to illustrate this. Suppose we wish to determine the service rate for each server so as to maximize the total throughput. Initially each server may have (approximately) equal impact on the system performance. If the service rate of some servers have been raised to a very high level, clearly a further increase of the service rate for these servers will result in very limited contributions to the system performance compared to other servers whose service rates are still at low level. In other words, whether a factor is influential or not can depend on the level that it is evaluated.

The rest of this paper is organized as follows. The main framework of STRONG-LS and the sequential screening designs will be presented in Section 2, followed by the numerical evaluations based on various test problems in Section 3. In Section 4, we apply STRONG-LS to solve an empirical problem to validate its viability in practical settings. We conclude with future research in Section 5.

2 EXTENDING STRONG FOR LARGE-SCALE PROBLEMS

2.1 Main Framework

In this section, we first provide a high-level introduction for STRONG, followed by a detailed account about how the screening designs are incorporated to enable STRONG to handle large-scale problems. The introduction of STRONG closely followed the steps presented in Chang, Hong, and Wan (2010). The main framework of STRONG is illustrated in Figure 1. STRONG is a metamodel-based framework built on the basis of efficient experimental design and regression analysis. There are two stages and an *inner loop* in STRONG. In Stage I, which is usually employed at early iterations when the algorithm is far from the optimum, STRONG constructs and optimizes a first-order model; in Stage II, which typically occurs when the first-order model is insufficient to represent the response surface, STRONG constructs and optimizes a second-order model. In both Stages I and II, the algorithm only uses a fixed number of observations to construct the models. However, if the constructed model cannot yield a satisfactory solution in Stage II, the algorithm will enter an *inner loop* where more simulation efforts are allocated to construct better second-order models until a good solution is found.

In traditional RSM, experimenters are required to iteratively specify the “region of interest” where a proper experimental design (also selected by experimenters) is employed. STRONG introduces the concept of "trust region" to replace the "region of interest" in traditional RSM. This introduction eliminates the requirement for experimenters' involvements. A trust region at a solution x' with a radius $\Delta > 0$ is defined as $\mathscr{B}(x',\Delta) = \{x \in \mathbb{X}^p : \|x - x'\| \leq \Delta\}$, where $\|\cdot\|$ denotes the Euclidian norm. At any iteration of the algorithm, for example iteration k, the current solution is denoted as x_k and the trust-region radius as Δ_k. Then the trust region at iteration k is $\mathscr{B}(x_k,\Delta_k)$. The size of trust region will be automatically updated by the algorithm.

At any iteration of the algorithm, for example iteration k, STRONG conducts the following five steps:

Step 1. Identify the important factors by performing the factor screening method;
Step 2. Construct a local model $r_k(x)$ around the center point x_k;
Step 3. Solve $x_k^* \in \text{argmin}\{r_k(x) : x \in \mathscr{B}(x_k,\Delta_k)\}$;
Step 4. Simulate several observations at x_k^* and estimate $g(x_k^*)$;
Step 5. Examine the quality of x_k^* and update x_{k+1} and the size of trust region Δ_{k+1}.

In Stage I, a resolution III fractional factorial design is employed to build the local model that includes all factors, and as a consequence, there is no need to perform Step 1. In Stage II, however, because it is computationally prohibitive to build a full quadratic model for large-scale problems, the factor screening method is used to identify the important factors. The selected factors are then used to develop the local model, followed by solving the new solution that can optimize the local model within the trust region. The algorithm examines the quality of the new solution. If the new solution is satisfactory, the trust region enlarges, while when the quality of the new solution is poor or merely acceptable, the trust region shrinks or remains the same size. Thus, the size of trust region is an effective indicator that can evaluate how successful the previous iterations are. If the trust region is large, which means most of the previous iterations are successful, a first-order model may be sufficient to serve as the local model, and the algorithm will go with Stage I; otherwise, a more elaborate model, i.e., the second-order model, will be used for better fitting, and the algorithm will switch to Stage II.

If the algorithm cannot find a satisfactory solution in Stage II, the possible reasons are analyzed as follows:

Reason 1. Some important factors are not considered in the model;
Reason 2. The quality of local model is poor;
Reason 3. The response variable contains large noise;
Reason 4. The optimization step is too aggressive.

Based on these four reasons, the *inner loop* takes the following four strategies accordingly:

Strategy 1. Include more factors to build the local model;
Strategy 2. Increase the number of design points;
Strategy 3. Increase the replication of center runs;
Strategy 4. Shrink the trust region.

Among these four strategies, Strategy 1 will be discussed in detail in next section. As for Strategy 2, 3 and 4, they are employed in a similar way as the STRONG algorithm. Interested readers are referred to (Chang et al. 2007)(Chang 2008)(Chang et al. 2010). The *inner loop* is indeed effective. We can prove that under certain regularity assumptions the algorithm can always find a satisfactory solution, if it is not at a stationary point, described as Theorem 1.

Theorem 1 For any $x_k \in \mathbb{X}^p$ and given k, if $\|\nabla g(x_k)\| > 0$, the algorithm can always find a new satisfactory solution in iteration k.

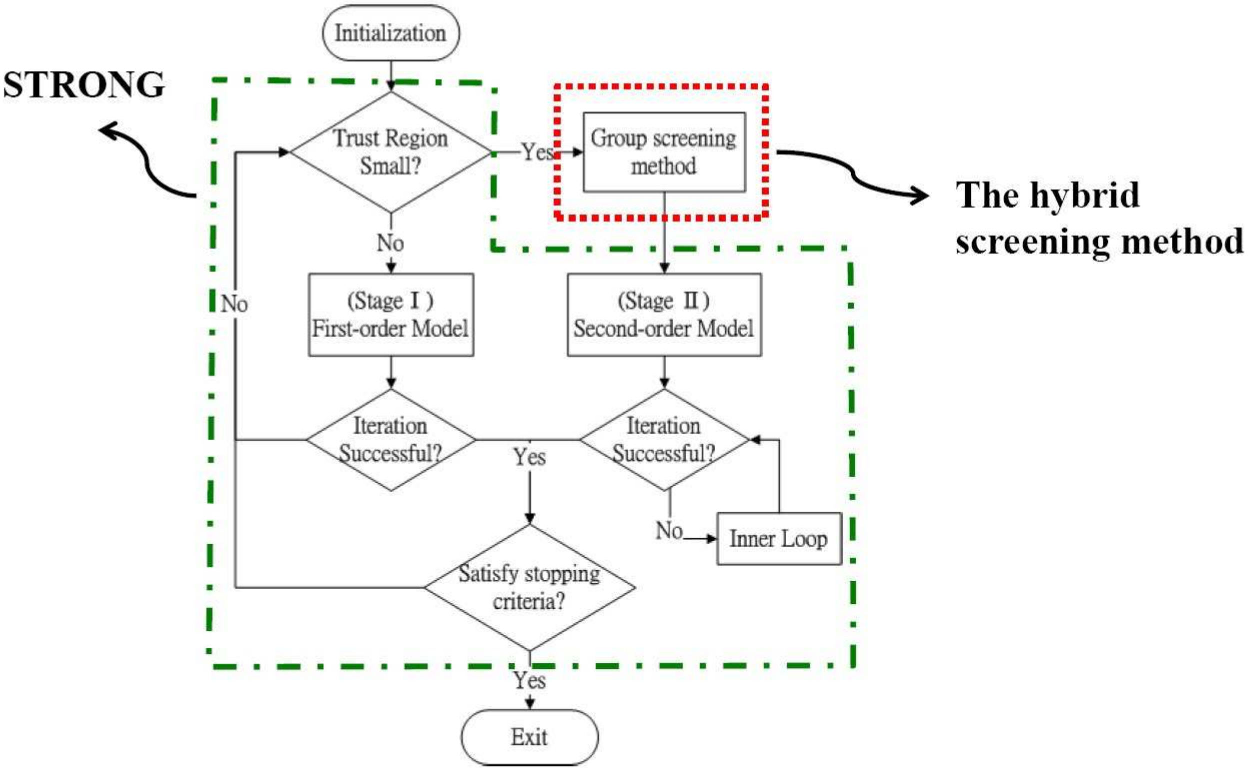

Figure 1: The main framework of STRONG-LS.

2.2 Sequential Screening Designs

The hybrid method is an efficient factor screening method that combines CSFD and CSB-X to identify the important factors for a large-scale stochastic system. The general framework of the hybrid method is presented in Figure 2. Specifically, there are two phases in the hybrid method. In Phase 1, the algorithm performs the prescreening procedure to obtain the estimate of each factor and determine whether CSB-X or CSFD will be used for further testing based on the effect estimates. In Phase 2, CSB-X and CSFD are performed separately based on the assignment of Phase 1. The prescreening stage not only eliminates the requirement for prior knowledge of the stochastic system, e.g., the sign of each factor effect, but also allows the subsequent screening procedures to perform in their favorable conditions, thereby resulting in a better overall efficiency compared to its component screening methods in general circumstances (Shen et al. 2010).

To perform the hybrid screening method, users are required to determine two values, Δ_0 and Δ_1. A factor is considered unimportant or important if its effect is less than Δ_0 or greater Δ_1. The error control of the hybrid method is stated as follows.

Theorem 2 (Shen et al. 2010) Given a prescreening method and a qualified testing procedure, the hybrid method guarantees that

$$\Pr\{\text{Declare effect } k \text{ important} |\, |\beta_k| \leq \Delta_0\} \leq \alpha,$$

and

$$\Pr\{\text{Declare effect } k \text{ important} |\, |\beta_k| \geq \Delta_1\} \geq (1-\zeta)\gamma,$$

for each effect where α and γ are the user-specified target Type I Error and power rates and ζ =Pr(a critical effect is not assigned to IMP group), which depends on the prescreening procedure.

Note that for the factors whose effect is between Δ_0 and Δ_1, it is also considered important but there is no error control over it. In the STRONG-LS algorithm, we propose to select the Δ_0 and Δ_1 in the following way:

$$\Delta_0 = \sum_{j=1}^{k} |\hat{\tau}_j| * s \text{ and } \Delta_1 = \text{mean}(|\hat{\tau}_j|) + \kappa_i * \text{std}(|\hat{\tau}_j|), \tag{2}$$

where τ_j represents the effect estimate of factor j, s is a constant and κ_i is a diminishing sequence satisfying $\lim_{i\to\infty} \kappa_i = 0$ (i refers to the counter of *inner loop*). Both s and κ_i are determined by users, for example, we can let $s = 0.001$ and $\kappa_i = 3 * 0.9^i$. When the algorithm just enters Stage II, all factors are put in the queue waiting to be determined whether it is an important factor, while in *inner loop* only the factors that are yet identified as important factors in the previous inner loops are considered. Therefore if the current local model cannot yield a satisfactory solution, more inner loops will be performed, κ_i decreases, and consequently more factors will be selected to build the local model. Under certain regularity conditions, we can prove that STRONG-LS can converge to the local optimal solution with probability one. We formally state this result in Theorem3.

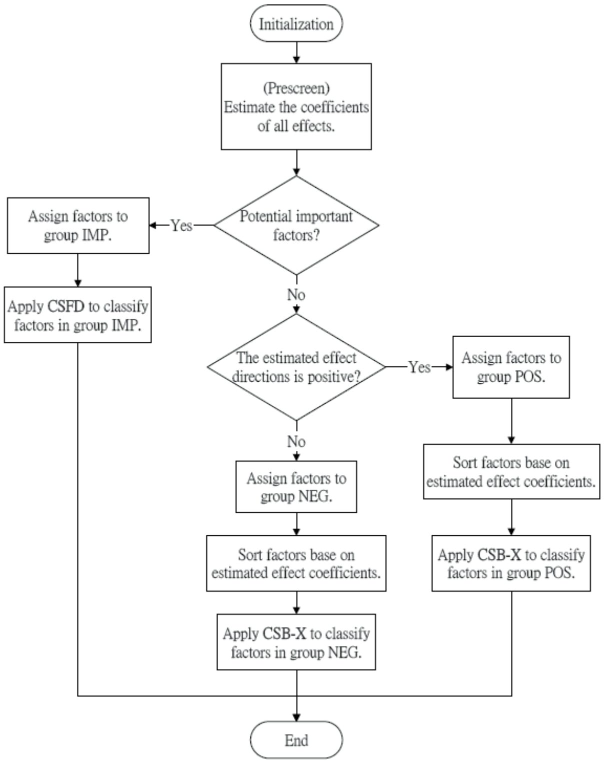
Initialization
(Prescreen) Estimate the coefficients of all effects.
Potential important factors?
Yes
Assign factors to group IMP.
Apply CSFD to classify factors in group IMP.
No
The estimated effect directions is positive?
Yes
Assign factors to group POS.
Sort factors base on estimated effect coefficients.
Apply CSB-X to classify factors in group POS.
No
Assign factors to group NEG.
Sort factors base on estimated effect coefficients.
Apply CSB-X to classify factors in group NEG.
End

Figure 2: The Hybrid Method.

Theorem 3 If STRONG-LS has infinitely many successful iterations, then $\liminf_{k\to\infty} \|\nabla g(x_k)\| = 0$ w.p.1.

3 NUMERICAL EVALUATIONS

In this section, we use several examples to illustrate the performance of the proposed STRONG algorithm. We benchmark with two competing algorithms: SPSA and the modified Nelder-Mead method (MNM). SPSA is widely used in academic community and is well-known for the advantage of only requiring two observations to estimate the gradient, regardless of the dimensionality of problem. MNM, on the other hand, is a direct-search-based method originally developed for deterministic nonlinear programming. Barton and Ivey (1996) proposed several modifications specifically designed for stochastic response variables.

To compare the performances of different algorithms, we use "Optimality Gap" (OG) defined below to evaluate the performance of algorithm:

$$\frac{g(x_k^*)-g(x^*)}{g(x_0)-g(x^*)},$$

where x_0, x^*, and x_k^* correspond to the initial solution, the true optima of the underlying response surface and the best solution before the algorithm terminates, respectively. If OG is close to 0, the algorithm is close to the optimum, and if OG is close to 1 or even greater than 1, the algorithm is far from optima or even diverge. We run 10 macroreplications for 6 scenarios for each algorithm (STRONG, MNM, SPSA) with fixed initial solutions $x_0 = 20 * 1_{200\times 1}$. For each macroreplication, the algorithm is terminated after $20,000$ observations are consumed.

3.1 Test Problems

We consider three types of problems, each of them as 200 decision variables. The first two types of problems are selected from literature in nonlinear optimization (More, Garbow, and Hillstrom 1981). Specifically, the first one is the classic Rosenbrock function,

$$g(x)=\sum_{i=1}^{p/2}[100(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2],$$

which is known to be a difficult problem even in deterministic settings. The second type is the Freudenstein and Roth function:

$$g(x)=\sum_{i=1}^{p/2}\;[-13+x_i+((5-x_{2i})x_{2i}-2)x_2]^2+[-29+x_i+((x_{2i}+1)x_{2i}-14)x_{2i}]^2,$$

which is a multimodal function (More, Garbow, and Hillstrom 1981). The third type is a quadratic function, i.e.,

$$g(x)=\sum_{i=1}^{p}x_i^2.$$

Montgomery (2009) stated that in real world "most systems are dominated by some of the main effects and low-order interactions, and most high-order interactions are negligible" (the so-called *sparsity of effects principle*). In order to assess the performance of STRONG when the *sparsity of effects principle* holds, two settings with respect to the percentage of important factors are considered: 5% and 100%. The 100 % setting refers to the original test function without any change. For the 5% setting, we keep the first 5% variables in the test function and multiply the rest 95% by 0 so they no longer affect the system performance.

The stochastic responses of all test problems take the form of $G(x) = g(x)+\varepsilon(x)$, where $\varepsilon(x) \sim$ Normal $(0,\sigma^2(x))$. The variance configuration is set as $\sigma(x) = 0.01*g(x)$, where $\sigma(x)$ corresponds to the standard deviation of noise. Notice that this setting can lead to very large variance of response variable when the initial solution is selected far away from the optimum (i.e., $g(x)$ is large). The stability of algorithm can therefore be tested.

The results are summarized in Tables 1. Given 10 macroreplications, 20,000 observations for each algorithm, the average OG is reported, along with the associated standard deviation given in the parentheses. If at least one replication fails to converge (i.e., OG$\geq$ 1), the ratio of successful replications is given in parentheses.

From Table 1, STRONG is remarkably successful, giving results that are much better than the other two algorithms in most of the test problems. To understand the computational advantage of STRONG

Table 1: Numerical comparison between STRONG, MNM, and SPSA.

	Percentage of Important Factors (5%)		
	Rosenbrock	Quadratic	Freudenstein and Roth
STRONG	1.92E-05(3.57E-21)	5.46E-08 (E-23)	4.01E-06 (8.93E-22)
MNM	0.77(9.56E-03)	0.91(0.01)	0.75(1.17E-02)
SPSA	(0%)	4.0E-11(1.41E-11)	(20%)

	Percentage of Important Factors (100%)		
	Rosenbrock	Quadratic	Freudenstein and Roth
STRONG	4.08E-05(7.14E-21)	3.50E-08 (1.06E-05)	4.47E-06 (2.87E-07)
MNM	0.98(3.75E-03)	(0%)	0.97(4.50E-02)
SPSA	(0%)	6.59E-08(3.31E-08)	(0%)

when the *sparsity of effects principle* holds, we further compare the numerical performance of STRONG when the system has 5% and 100% important factors respectively.

In Figure 3, the red solid line refers to the 5% setting while the green broken line refers to the 100% setting. We found that STRONG enjoys significant computation gains in the 5% setting in all three test functions. This is attributed to the effectiveness of sequential screening design, which coincides with the finding in Wan et al. (2006) where they reported that the sequential screening design is particularly efficient when the percentage of important factors is small.

4 EMPIRICAL APPLICATION

In this section, we apply the developed STRONG algorithm to solve a two-stage stochastic programming problem. Given in Shapiro et al. (2009) p.9, this problem is concerned with the optimal ordering decision for a multiproduct assembly manufacturer. Suppose the manufacturer produces n products. There are m different parts that have to be ordered from third-party suppliers. A unit of product i requires a_{ij} units of part j, where $i = 1,\dots,n$ and $j = 1,\dots,m$. The demand for the products is a random vector $D = (D_1,\dots,D_n)$. Before the demand is materialized, the manufacturer needs to preorder the parts from the supplier at a cost of c_j per unit of part j. After the demand is observed, the manufacturer may decide which portion of the demand is to be satisfied, so that the available numbers of parts will not be exceeded. It costs additionally l_i to satisfy a unit of demand for product i, and the unit selling price of this product is q_i. The parts not used are assessed salvage values $s_j < c_j$ and the unsatisfied demand is lost.

Let the number of parts ordered be $x_j,\ j = 1,\dots m$. After the demand D becomes known, we need to determine how much of each product to make. Let us denote the numbers of units produced by $z_i,\ i = 1,\dots,n$ and the numbers of parts left in inventory by $y_j,\ j = 1,\dots,m$.

When the demand $d = (d_1,\dots,d_n)$ is known, the best production plan can be obtained by solving the following linear programming problem (expressed in matrix form):

$$\begin{aligned} &\min_{z,y} (l-q)^T z - s^T y \\ &\text{s.t. } y = x - A^T z, \\ &0 \le z \le d,\ y \ge 0. \end{aligned} \tag{3}$$

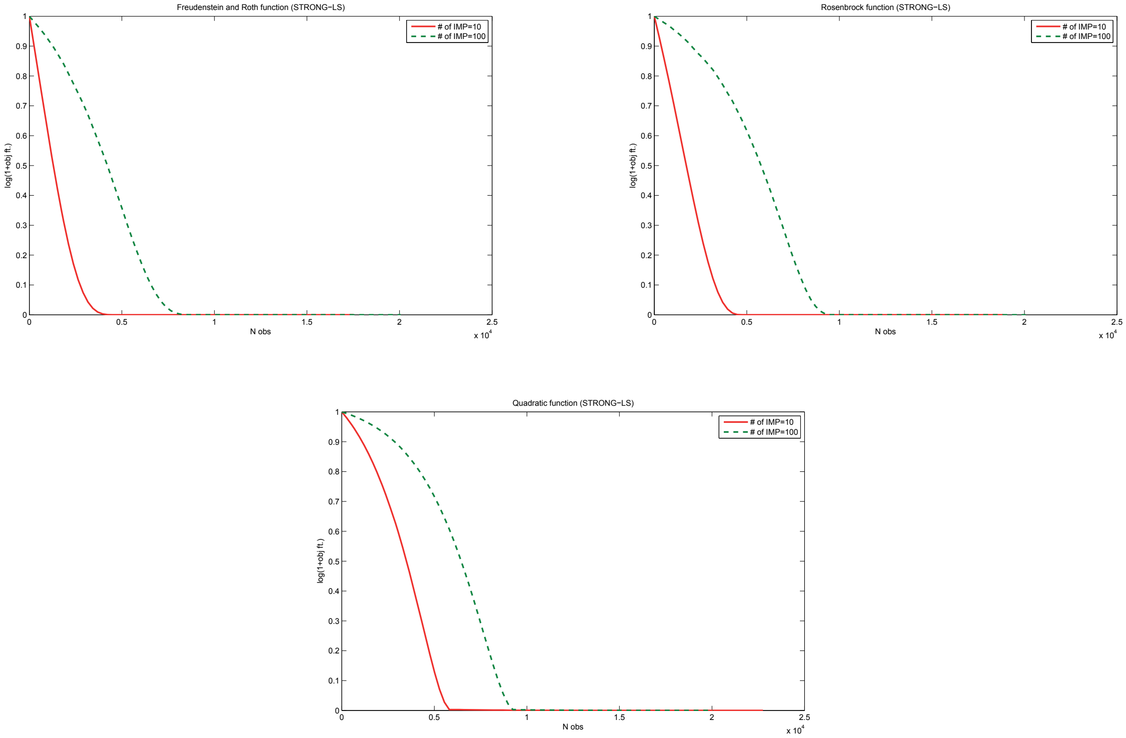

Figure 3: Computational comparison of STRONG in small and large percentage of important factors.

Let $Q(x,D)$ denote the optimal value of problem (3). The quantities of parts to be ordered can be determined from the following optimization problem.

$$\min_{x\geq 0} c^T x + \mathrm{E}\left[Q(x,D)\right], \tag{4}$$

where the expectation is taken with respect to the probability distribution od the random demand vector D.

Note that $\mathrm{E}\left[Q(x,D)\right]$ does not have a closed-form expression and can only be treated as a block box. The stochastic response $Q(x,D)$ can be obtained by generating a random demand vector D and then solving the linear programming problem. The decision variable x that represents the quantity of parts to be ordered is typically of large scale in practice. As a result, solving Problem (4) is very challenging.

Based on the parameter settings given in Table 2, we apply three algorithms to solve the problem: STRONG, MNM and SPSA, and terminate the algorithm when 10,000 observations are consumed. Note that the true optimal solution of this problem is not available; therefore, instead of reporting the OG, we report the trajectory of the (estimated) objective value vs. the number of observations consumed. Suppose the demand for each product is uniformly distributed in (0, 5). The simulation results show that SPSA cannot converge to any solution in this problem. Therefore, we only report the trajectory of STRONG and MNM in Figure 4. It can be easily seen that STRONG has significant improvements in the objective function within 8,000 observations while MNM only make very limited progress when the algorithm is terminated.

Table 2: Parameter settings.

Parameter	m	n	a_{ij}	c_j	d_j	l_i	q_i	s_j
Setting	200	100	$\sim \mathrm{Unif}(0,10)$	50	$\sim \mathrm{Unif}(0,5)$	$\sim \mathrm{Unif}(20,30)$	$\sim \mathrm{Unif}(200,300)$	5

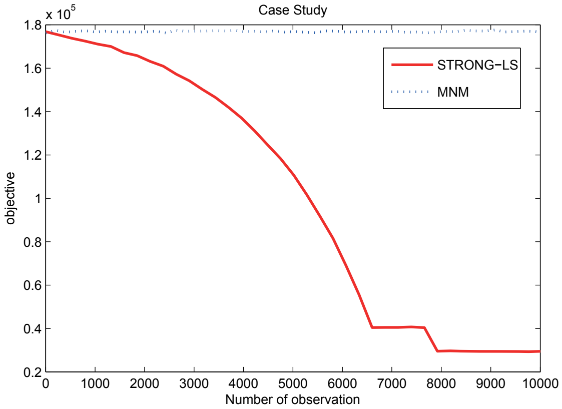

Figure 4: The performance of STRONG and MNM when solving the empirical problem.

5 CONCLUDING REMARKS

In this paper, we exploited the structure of STRONG and combined it with efficient screening designs to cope with large-scale simulation optimization problems. The newly-developed algorithm STRONG sequentially selects a subset of important factors to build the local surface and performs optimization on these selected factors for improving the objective function. Because the large-scale problem is broken into several small ones, the problem becomes much easier to solve. It can be proved that STRONG can converge to the local optimum of the original problem. The numerical evaluations show that STRONG can outperform the competing algorithms, MNM and SPSA, in both the unimodal and multi-modal response surfaces. The empirical application also validates the viability of STRONG in practical settings.

The future research will include the extension of STRONG to handle the constrained problems. Moreover, we will investigate the adaption of STRONG for problems where the gradient estimator is available to improve its computational efficiency.

ACKNOWLEDGMENT

This research was partially supported by National Science Council in Taiwan (NSC98-2218-E-007-015).

REFERENCES

Barton, R., and J. Ivey. 1996. “Nelder-Mead simplex modifications for simulation optimization”. *Management Science* 42 (7): 954–973.

Chang, K.-H. 2008, December. *Stochastic Trust Region Response Surface Convergent Method for Continuous Simulation Optimization*. Ph. D. thesis, Purdue University, West Lafayette, Indiana.

Chang, K.-H., L. Hong, and H. Wan. 2007, December. “Stochastic Trust Region Gradient-Free Method (STRONG)-A New Response-Surfance-based Algorithm in Simulation Optimization”. In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 346–354. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chang, K.-H., L. Hong, and H. Wan. 2010. “Stochastic Trust-Region Response-Surface Method (STRONG)-A New Response-Surfance Framework for Simulation Optimization”. *working paper*.

Chang, K.-H., and H. Wan. 2009, December. “Stochastic Trust Region Response Surface Convergent Method for Generally-Distributed Response Surface”. In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 563–573. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Conn, A., N. Gould, and P. Toint. 2000. *Trust-Region Methods*. SIAM.

Fu, M. 2002. “Optimization for Simulation: Theory vs. Practice”. *INFORMS Journal on Computing* 14:192–227.

Kushner, H., and G. Yin. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer.

Montgomery, D. 2009. *Design and Analysis of Experiments*. 7 ed. New York: John Wiley and Sons.

More, J., B. Garbow, and K. Hillstrom. 1981. “Testing Unconstrained Optimization Software”. *ACM Transactions on Mathematical Software* 7:17–41.

Morris, M. 2006. “An overview of group factor screening”. In *Screening: Methods for experimentation in industry*, edited by A. Dean and S. Lewis, 191–206. New York: Springer-Verlag.

Myers, R., D. Montgomery, and C. Anderson-Cook. 2009. *Response Surface Methodology-Process and Product Optimization Using Designed Experiments*. New York: John Wiley and Sons.

Robbins, H., and S. Monro. 1951. “A stochastoc approximation method”. *The Annals of Mathematical Statistics* 22:400–407.

Sanchez, S., H. Wan, and T. Lucas. 2009. “Two-phase screening procedures for simulation experiments”. *ACM Transactions on Modeling and Computer Simulation* 19 (2): 1–24.

Shapiro, A., D. Dentcheva, and A. Ruszczyński. 2009. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM-Society for Industrial and Applied Mathematics.

Shen, H., and H. Wan. 2009. “Controlled sequential factorial design for simulation factor screening”. *European Journal of Operational Research* 198:511–519.

Shen, H., H. Wan, and S. Sanchez. 2010. “A Hybrid Method for Simulation Factor Screening”. *Naval Research Logistics* 57:45–57.

Spall, J. 2003. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. New York: John Wiley and Sons.

Trocine, L., and L. Malone. 2000, December. “Finding important independent variables through screening designs: A comparison of methods”. In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 749–754. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Trocine, L., and L. Malone. 2001, December. “An overview of newer, advanced screening methods for the initial phase in an experimental design”. In *Proceedings of the 2001 Winter Simulation Conference*, edited by B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 169–178. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wan, H., B. Ankeman, and B. Nelson. 2010. “Improving the Efficiency and Efficacy of Controlled Sequential Bifurcation for Simulation Factor Screening”. *INFORMS Journal on Computing* 22 (3): 482–492.

Wan, H., B. Ankenman, and B. Nelson. 2006. “Controlled Sequential Bifurcation: A New Factor-Screening Method for Discrete-Event Simulation”. *Operations Research* 54:743–755.

AUTHOR BIOGRAPHY

KUO-HAO CHANG is an assistant professor in Industrial Engineering and Engineering Management at National Tsing Hua University. He received his PhD in Industrial Engineering from Purdue University. His research interests include simulation optimization, stochastic models and Monte Carlo simulation. His email address is chang@mx.nthu.edu.tw.

MING-KAI LI is a mater student in National Tsing Hua University. He will join Taiwan Semiconductor Manufacturing Company (TSMC) after his graduation this year. His research interests include large-scale simulation optimization, and simulation applications in semiconductor manufacturing. His e-mail address is s9834525@gmail.com.

HONG WAN is an associate professor in the School of Industrial Engineering at Purdue University. Her research interests include design and analysis of simulation experiments, simulation optimization;

simulation of manufacturing, healthcare and financial systems; quality control and applied statistics. She has taught a variety of courses and is a member of INFORMS and ASA. Her e-mail and web addresses are hwan@purdue.edu and http://web.ics.purdue.edu/~hwan/index.htm.