Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

STOCHASTIC TRUST REGION GRADIENT-FREE METHOD (STRONG) - A NEW RESPONSE-SURFACE-BASED ALGORITHM IN SIMULATION OPTIMIZATION

Kuo-Hao Chang

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907-2023, U.S.A.

L. Jeff Hong

Dept. of Industrial Engineering and Logistics Management
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, China

Hong Wan

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907-2023, U.S.A.

ABSTRACT

Response Surface Methodology (RSM) is a metamodel-based optimization method. Its strategy is to explore small subregions of the parameter space in succession instead of attempting to explore the entire parameter space directly. This method has been widely used in simulation optimization. However, RSM has two significant shortcomings: Firstly, it is not automated. Human involvements are usually required in the search process. Secondly, RSM is heuristic without convergence guarantee. This paper proposes Stochastic Trust Region Gradient-Free Method (STRONG) for simulation optimization with continuous decision variables to solve these two problems. STRONG combines the traditional RSM framework with the trust region method for deterministic optimization to achieve convergence property and eliminate the requirement of human involvement. Combined with appropriate experimental designs and specifically efficient screening experiments, STRONG has the potential of solving high-dimensional problems efficiently.

1 INTRODUCTION

This paper proposes a novel design-of-experiment (DOE) based simulation optimization algorithm. Simulation Optimization is defined as "a repeated analysis of the simulation model with different values of design parameters, in an attempt to identify best simulated system performance" (Barton and Meckesheimer 2006). The response is assumed to be a black-box, and can only be evaluated by computer simulations. This paper focuss on the problems with continuous decision variables. The challenges arise in the presence of randomness of the response function and the unknown relationship between response and decision variables which contrasts with classic deterministic optimization assuming that users has perfect information about the objective function including gradient function and Hessian matrix.

Consider the following minimization problem

$$\min_{x\in\mathbb{R}^p} E(G(x,\omega)) \tag{1}$$

$G(x,\omega)$ is the simulation output and $G(x,\omega) = g(x) + \varepsilon_x$. Here $g(x)$ is an unknown underlying relationship between response and decision variables; ω represents the random variable defined in the probability space $(\Omega, \mathscr{F}, \mathscr{P})$; and ε_x is the randomness induced by ω. Function $G(x,\omega)$ is assumed measurable. Notice that the subscript x in ε_x shows the dependence of the randomness on its location. This dependence is indeed realistic in broad applications (Spall 2003). We further assume that:

1. $\varepsilon_x \sim N(0,\sigma_x^2)$, and
2. σ_x^2 is unknown and $\sup_x \sigma_x^2 < \infty$

Given an oracle which is capable of generating the response $G(x,\omega)$, our goal is to find the stationary point of the underlying response function $g(x)$.

Simulation optimization methodologies can be generally classified into three categories, stochastic approximation (SA), sample path optimization (SPO), and response surface methodology (RSM) (Fu 2002). The method of stochastic approximation attempts to imitate the deterministic gradient search method (e.g., steepest descent method) (Fu 2002). It is a iterative process of moving from one solution to another based on the gradient estimate. Frequently used gradient estimation methods are, for example, finite difference estimation, likelihood ratio/score function method, perturbation analysis and frequency domain method (Carson and Maria 1997). SA has rich convergence theories in various conditions, however, it usually suffers from slow convergence

rate when the objective function is flat and often diverges when the objective is steep (Andradottir 1995).

SPO uses the average function to approximate the original objective function and then optimize it (average function) by applying a deterministic optimization method. Generally, in order to make the original objective function well-approximated, a large number of sample paths are required for the average function. One advantage of the the sample path approach is that it can "bear the large and powerful array of deterministic optimization methods and deal with problems in which the parameters might be subject to complicated constraints" (Robinson 1996).

Response surface methodology (RSM) is a sequential procedure for building relationship (polynomial functions) between the input and output variables and optimizing this polynomial functions. The success of RSM is based on the validity of repeated localized curve fitting and subsequent optimization (Spall 2003). RSM has an arsenal of powerful statistical tools including regression and analysis of variance at its disposal (Fu 1994). Furthermore, combining with efficient experimental designs, it has the potential to solve high-dimensional problems. However, RSM has two main flaws. First, RSM uses steepest descent (or ascend) method (SD) to search for the next improved region. SD suffers from two well-known problems (Myers and Montgomery 2002): (i) it is scale-dependent; (ii) the step size along its path is selected intuitively; human involvements are required in each iteration. Second, RSM provides no convergence guarantee, i.e., even if it is kept running for long time there is no assurance that the solution will be close to the optimal solution. Some research has been conducted to provide an automated version of RSM (Nicolai, Dekker, Piersma, and Oortmarssen 2004). However, there is little research addressing the convergence issue as the best of our knowledge.

Trust Region Method (TR) is a well-studied deterministic nonlinear programming method (Nocedal and Wright 1999) that has similar framework as response surface methodology. At each iteration, TR defines a trust region around the current iterate, build a quadratic approximation model, find the optimal solution within the trust region (based on this quadratic model) and then moves to the optimal solution (if it is satisfactory by certain criterion). The size of trust region will be automatically determined by the algorithm. In addition, it can be proved to converge to the stationary point of the objective function.

However, TR is developed for deterministic optimization, (i.e., the objective function is without noise and can be expressed explicitly) therefore it cannot handle stochastic case. Besides, TR builds the quadratic model all the time therefore it could be computationally intensive when the dimensionality of problem grows. This research proposed an algorithm called STRONG which combines RSM and TR. STRONG takes the benefits from both of them so that it can handle stochastic optimization problems, requires no human involvements, and have convergence guarantee. Furthermore, it is capable of solving high-dimensional problems efficiently if combined with efficient experimental designs. This article will be organized as follows. In section 2, we will review response surface methodology and trust region method. In section 3, the proposed algorithm STRONG will be introduced and its performance is discussed. Section 4 will demonstrate preliminary numerical results. We conclude with the future direction in section 5.

2 RESPONSE SURFACE METHODOLOGY AND TRUST REGION METHOD

2.1 Response Surface Method

Response surface methodology (RSM) introduced in the early 50's by Box and Wilson (1951) is a powerful tool to determine the optimal input combination. Classic RSM employs a sequential experimental procedure to fit and optimize a series of polynomials on small subregions. RSM has been extensively used in industry to select the optimal operating conditions or product designs. A vast amount of literatures on RSM can be found, for example, (Box and Wilson 1951), (Box, Hunter, and Hunter 1978), (Myers and Montgomery 2002), (Khuri and Cornell 1996).

One of the earliest applications of RSM in simulation was reported by Biles (1974) and Kleijnen (1975). Since the simulation model representing the real world systems can be very complex, difficult and expensive to construct, some simple approximation models (sometimes called metamodels or surrogate models) such as linear or quadratic local models will be locally constructed to "mimic" the behavior in the subregion. These metamodels reduce the prediction variance by extending the effect of the law of large numbers over all design points in the local region (Barton and Meckesheimer 2006). RSM is one of the metamodel-based methods. The fundamental RSM framework can be briefly described as follows (Myers and Montgomery 2002):

Stage I

1. Plan and run a factorial (or fractional factorial) design plus the center points in the region of interest.
2. Do the curvature test. If the curvature test is not significant, go to step 3. Otherwise go to Stage II.
3. Fit a linear regression model to the data.
4. Determine the path of steepest descent (if it is a minimization problem).
5. Move along the path of steepest descent until the response no longer improves.

(Stage I will be repeated until the the linear response surface becomes inadequate.)

Stage II

1. Run a central composite design and fit a quadratic model.
2. Based on the quadratic model, find the optimal solution.

Based on the framework above, RSM sequentially explores local regions of the whole experimental region and performs line searches to approach the optimal region. Its moving direction is chosen by the steepest descent (which is obtained from the local model) and step size is determined by a line search. It is notable that Phase II is only done once, whereas Stage I is iterated a number of times. Fewer replications are expended for each iteration in Stage I while in Phase II the region should be explored thoroughly by using a large number of replications (Fu 1994). Notice that the optimal solution is obtained through the quadratic model in Stage II, therefore it is an "estimated" optimal solution and will not necessarily be a true optimal solution for the underlying response.

One of the biggest advantage in RSM is its generality (Fu 1994). An arsenal of well-studied statistical tools such as regression analysis, design of experiments and ANOVA can be incorporated in its framework. However, several issues need to be solved. Firstly, RSM is not automated. Human opinions are required to define the region of interest and to decide the experimental design for each iteration. Secondly, RSM is heuristic and does not possess convergence property. Convergence is an important issue because it can provide valuable insights of the algorithm, e.g., to delineate the range of applicability and limitations of the algorithm.

2.2 Trust Region Method

Trust Region Method (TR) is a nonlinear optimization algorithm for deterministic problems. Consider a minimization problem

$$\min_{x\in\mathbb{R}^{\kappa}} f(x) \tag{2}$$

where $f(x)$ is an explicit deterministic function. A basic framework of trust region method is given as follows (Conn, Gould, and Toint 2000):

1. Given an initial point x_0, an initial trust region radius $\triangle_0$, and constants η_1, η_2, γ_1, γ_2 satisfying $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_1 \leq \gamma_2 < 1$. Compute $f(x_0)$ and set k=0
2. Choose $\|\cdot\|_k$ and define a local model m_k (usually by Taylor expansions) in $B_k = \{x \in \mathbb{R}^n | \ \|x - x_k\|_k \leq \triangle_k\}$
3. Compute a step s_k that "sufficiently reduces" the model m_k and $x_k + s_k \in B_k$
4. Compute $f(x_k + s_k)$

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

If $\rho_k > \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise define $x_{k+1} = x_k$

5. Set

$$\triangle_{k+1} \in \begin{cases} [\triangle_k, \infty) & \text{if} \quad \rho_k \geq \eta_2, \\ [\gamma_2\triangle_k, \triangle_k] & \text{if} \quad \rho_k \in [\eta_1, \eta_2) \\ [\gamma_1\triangle_k, \gamma_2\triangle_k] & \text{if} \quad \rho_k < \eta_1. \end{cases}$$

Increase k by 1 and go to step 1.

A significant strength of TR is its convergence property. By convergence we mean the assurance that starting at an arbitrary initial solution will converge to a stationary point for the original problem. It is worth noting that since trust region method is used in deterministic optimization, the objective function is assumed available and can be expressed explicitly. The gradient and Hessian information are also known for each x in the domain. Therefore, the local model m_k can be constructed analytically. However, in stochastic setting, both the objective function, the gradient and Hessian matrix of each x in the domain are not known and can only be obtained by "estimation". In section 3, we will show how to modify the trust region framework so that it can be used in the stochastic setting.

2.3 Comparisons of RSM and TR

RSM and TR share some similarities. They both build the local linear/quadratic model for each iteration and use it to generate the search direction. However, there exists some difference between them.

1. TR cannot be used in the stochastic problems.
2. RSM and TR both generate steps based on the local model, but they use the model in different fashion. RSM uses the local model to generate a search direction first and then determines a suitable step length along this direction while trust region method defines a "trust region" around the current solution and chooses an optimal solution of the local model within the trust region. Therefore, the search direction and the step size of TR will be determined simultaneously (Nocedal and Wright 1999).
3. RSM starts with linear models and constructs a quadratic model only in the last step. TR, on the other hand, always use the quadratic model for each iteration so it can be computational intensive when the dimensionality grows.

4. RSM is a heuristic method without convergence guarantee while TR provides convergence guarantee.
5. RSM requires human involvements but TR is an automated algorithm.

Since RSM and TR both have their advantages and disadvantages, we proposed a method called STRONG combining the two methods together. STRONG not only can handle the stochastic case like RSM but also have convergence property like TR. We will introduce STRONG in section 3.

3 STOCHASTIC TRUST REGION GRADIENT-FREE METHOD

We proposed a response-surface-based algorithm called **STRONG** (**S**tochastic **TR**ust Regi**ON** **G**radient-Free Method) to solve the unconstrained stochastic optimization problems. "Gradient-Free" here means that users don't have any explicit information of gradient, namely $\frac{\partial G}{\partial x}$, or implicit gradient function. Rather, users can only obtain the gradient information (generally noisy) by estimation. STRONG consists of two algorithms, main algorithm and sub-algorithm. The main algorithm is the major framework which approaches an "improved" region while the sub-algorithm is to find a "satisfactory" solution within the trust region. Following are assumptions for the underlying response function and the local approximation model:

Assumption 1. *The underlying response function $g(x)$ is bounded below in $\mathbb{R}^n$.*

Assumption 2. *The underlying response function $g(x)$ is twice differentiable and its gradient and Hessian matrix are uniformly bounded in $\mathbb{R}^n$, i.e., $\exists\ \alpha_1, \beta_1$ s.t.*
$\| \nabla g(x) \| \leq \alpha_1,\ \| H(x) \| \leq \beta_1\ \ \forall x$

Assumption 1 assures that the underlying response function is bounded below. Therefore, there is at least a local minimal solution. Assumption 2 provides the assurance that Taylor expansion can work for every x in the parameter space. Assumption 3 makes sure that every estimated local model is uniformly bounded.

3.1 Metamodel Construction

We first give the following definitions which is relevant to STRONG.

Definition 1. *The trust region is the set of all points $\mathscr{T}^k = \{x \in \mathbb{R}^p | \ \| x - x^k \| \leq \triangle_T^k\}$*

Definition 2. *The sampling region is the set of all points $\mathscr{S}^k = \{x \in \mathbb{R}^p | \ \| x - x^k \| \leq \triangle_S^k\}$*

Definition 3. $\overline{G^N(x)} = \frac{\sum_{i=1}^N G_i(x,\omega)}{N}$

Definition 4. *X_n denotes the orthogonal design matrix*

$$X_n = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

Definition 5. *$Y_{n\times 1}$ denotes the revised response matrix in which each element subtracts the average of response value of the center point*

$$Y_{n\times 1} = \begin{bmatrix} y_1 - \overline{G^N(x)} \\ y_2 - \overline{G^N(x)} \\ \vdots \\ y_n - \overline{G^N(x)} \end{bmatrix}$$

Without loss of generality, we consider the following quadratic metamodel. If the metamodel is linear then the quadratic term will be ignored. STRONG requires orthogonality for the main-effects experimental design.

$$\gamma(d) = \widehat{g}(x) + \widehat{\nabla} g(x) d + \frac{1}{2} d^T \widehat{H}(x^k) d \tag{3}$$

where

1. $\widehat{g}(x) \triangleq \overline{G^N(x)}$
2. $\widehat{\nabla} g(x) \triangleq (X_n^T X_n)^{-1} X_n^T Y_n$

Notice that $\widehat{H}(x^k)$ can also be estimated if X_n is augmented up to the quadratic terms. It is worth mentioning that the gradient $\widehat{g}(x)$ in STRONG is not obtained by the typical Ordinary Least Squares (OLS) method. In STRONG, the intercept is the average of observations at point x and only the gradient and Hessian matrix are estimated by OLS. This allows us to prove the properties:

$$\lim_{N\to\infty} \widehat{g}(x) = g(x) \quad w.p.1 \tag{4}$$

$$\lim_{n\to\infty} \widehat{\nabla} g(x) = \nabla g(x) \quad w.p.1 \tag{5}$$

3.2 Hypothesis Testing Scheme

The following hypothesis testing procedure will guarantee that the algorithm will make a "substantial" move from x^k to x^{k+1} in which the new iterate solution can yield a sufficient reduction. Here k denotes the iteration counter in the main algorithm and k_i denotes the iteration counter in the sub-algorithm.

Definition 6. *x^k yields "sufficient reduction" if* $g(x^k) - g(x^{k+1}) \geq \frac{1}{2} \parallel \widehat{\nabla} g(x^k) \parallel min(\frac{\parallel \widehat{\nabla} g(x^k) \parallel}{\parallel \widehat{H}(x^k) \parallel}, \triangle_T^k)$

Hypothesis testing
H_0: x^{k+1} cannot yield sufficient reduction
H_1: x^{k+1} can yield sufficient reduction
if $t = \frac{\sqrt{N}(\overline{G^N(x^k)} - \overline{G^N(x^{k+1})})}{s} > t_{1-\alpha_k}$ then reject H_0, i.e., we claim that x^{k+1} can yield sufficient reductions. The significance level for iteration k is specified as α_k, which represents the probability that x^{k+1} cannot yield sufficient reduction while we accept it. The α_k is predesigned by the users but needs to satisfy the following requirement:

$$\sum_{k=1}^{\infty} \alpha_k < \infty \tag{6}$$

3.3 Main Algorithm

The following definitions will be used in STRONG algorithm.

Definition 7. *Ψ^{k_i} denotes the reject-solution set which collects all the visited solutions up to k_i in sub-algorithm* $\Psi^{k_i} = \{x^{k_1}, x^{k_2}, x^{k_3}, ..., x^{k_i}\}$

Definition 8. *Subproblem*
$\min_{d^k \in \mathbb{R}^p} \gamma^k(d^k) = \widehat{g}(x^k) + \widehat{\nabla} g(x^k) d^k + \frac{1}{2} d^{kT} \widehat{H}(x^k) d^k$
s.t. $\parallel d^k \parallel \leq \triangle_T^k$

Definition 9. *Iteration k(k_i) is called "successful" in the main algorithm (sub-algorithm) if $\rho^k > \eta_0$ ($\rho^{k_i} > \eta_0$)*

Main Algorithm

Stage I

1. Initialization: Specify an initial point x^0, initial trust region and sampling region parameters satisfying $0 < \triangle_S^0 < \triangle_T^0$, sample size n, N ($N > n$) and some constants $\eta_0, \eta_1, \gamma_1, \gamma_2$ satisfying $0 < \eta_0 < \eta_1 < 1$, $0 < \gamma_1 < 1 < \gamma_2$. Set k=0
2. Employ a proper experimental design with the center points. Take n replications for each design point and N replications for the center point in the sampling region.
3. Fit a linear local model.
4. Determine the path of steepest descent.
5. Perform the line search to find the next solution x^{k+1} within the trust region.
6. "Reduction test": If $\overline{G^N(x^k)} \leq \overline{G^N(x^{k+1})}$, increment k by 1, move to x^{k+1} and go back to step 2. Otherwise go to Stage II.

Stage II

1. Employ a proper experimental design.
2. Fit a quadratic model $\gamma^k(x)$
3. Find an approximate solution to the subproblem.
   Step1:
   $d^k = \arg\min_{d \in \mathbb{R}^p} \widehat{g}(x^k) + \widehat{\nabla} g(x^k) d \;\; s.t. \parallel d \parallel \leq \triangle_T^k$
   Step2:
   $\tau^k = \arg\min_{\tau > 0} \gamma^k(\tau d^k) s.t. \parallel \tau d^k \parallel \leq \triangle_T^k$
   Set $d_C^k = \tau^k d^k$, $x_C^{k+1} = x_C^k + d_C^k$
4. Acceptance/Rejection
   Compute $\rho^k = \frac{\overline{G^N(x_C^k)} - \overline{G^N(x_C^{k+1})}}{\gamma^k(0) - \gamma^k(d_C^k)}$
5. Trust Region Update
   if $\rho^k < \eta_0$, $\triangle_T^{k+1} = \gamma_1 \triangle_T^k$, $\triangle_S^{k+1} = \gamma_1 \triangle_S^k$. Let $x_C^k = x^{k_0}$and go to sub-algorithm.
   elseif $\eta_0 < \rho^k < \eta_1$, $\triangle_T^{k+1} = \triangle_T^k$, $\triangle_S^{k+1} = \triangle_S^k$, let $x^{k+1} = x^k + d_C^k$ go to step 6.
   else $\rho^k > \eta_1$ $\triangle_T^{k+1} = \gamma_2 \triangle_T^k$, $\triangle_S^{k+1} = \gamma_2 \triangle_S^k$, let $x^{k+1} = x^k + d_C^k$, go to step 6.
6. "Sufficient reduction" test: Do the sufficient reduction hypothesis test with type I error α_k (section 3.2). If the test is passed, increment k by 1, move to $x^{k+1}$and go to step 1. Otherwise, let $x_C^k = x^{k_0}$ and go to sub-algorithm.

Figure 1 illustrates the framework of main algorithm of STRONG. Notice that the selection of "proper experimental design" in stage I could be very flexible. For example, a fractional factorial design with center points is appropriate for stage I. In stage II, a proper experimental design could be an augmented orthogonal design such as central composite design so that a quadratic model can be fitted. In Stage I, STRONG is similar as RSM. It searches an "improved" region with the least computational effort. In stage II, Trust Region framework will replace the original RSM stage II. It is worth mentioning that STRONG will repeat the process of stage II until converge, whereas in RSM the stage II is just done once.

3.4 Sub-algorithm

When the local model constructed in the main algorithm cannot generate a satisfactory solution in STAGE, STRONG will initiate the sub-algorithm. Sub-algorithm will use simulation-allocation rule (SAR) (Hong and Nelson 2006) to allocate simulation observations for the reject-solution set. Let $n_{k_i}(x)$ be the additional observations allocated to x^{k_i} at iteration k_i as determined by the SAR. Then $N_{k_i}(x) = \sum_{j=0}^{i} a_j(x)$ denotes the total number of observations on solution x^{k_i} at iteration k_i for every $x \in \Psi^{k_i}$

Definition 10. *(Hong and Nelson 2006) The SAR guarantees that $a_j(x) \geq 1$ if x is a newly visited solution at iteration*

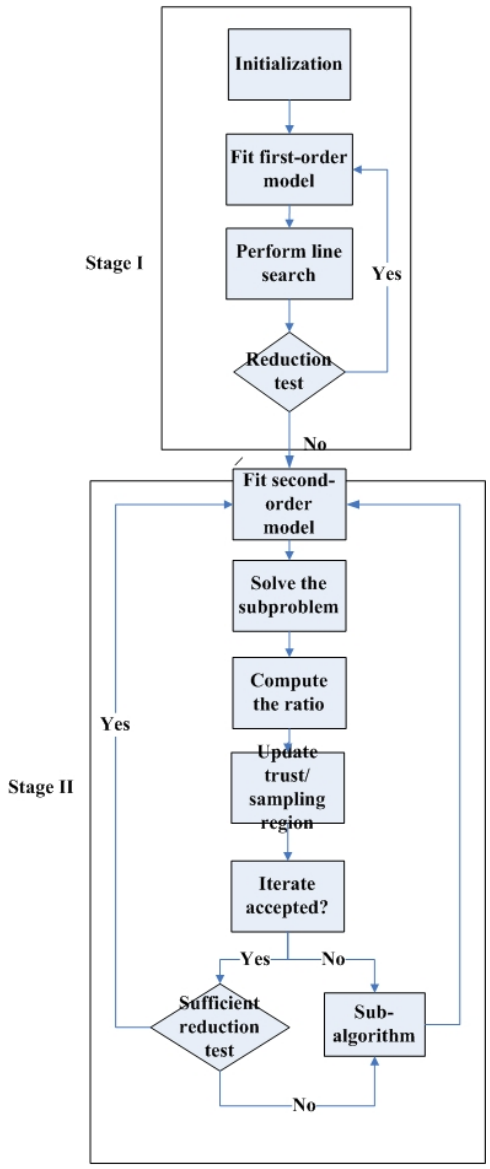

Figure 1: Framework of the main algorithm.

k_i $(x \in \Psi^{k_i} \setminus \Psi^{k_{i-1}})$ *and* $\lim_{i\to\infty} N_{k_i}(x) = \infty$ *for all visited solutions* $(x \in \bigcup_{k=0}^{\infty} \Psi^{k_i})$

Sub-algorithm

1. Employ a proper experimental design in the sampling region.
2. Use the accumulated design points from previous iterations to construct a quadratic approximation model γ^{k_i}
3. Find an approximate solution to the subproblem.
   Step 1:
   $d^{k_i} = \arg\min_{d\in\mathbb{R}^p} \widehat{g}(x^{k_i}) + \widehat{\nabla} g(x^{k_i})d \;\; s.t. \parallel d \parallel \leq \triangle_T^{k_i}$
   Step 2:
   $\tau^{k_i} = \arg\min_{\tau>0} \gamma^{k_i}(\tau d^{k_i}) s.t. \parallel \tau d^{k_i} \parallel \leq \triangle_T^{k_i}$
   Set $d_C^{k_i} = \tau^k d^{k_i}$, $x_C^{k_i+1} = x_C^{k_i} + d_C^{k_i}$. Determine $a_j(x)$ by SAR and update the reject-solution set $\Psi^{k_i} = \{x_C^{k_0}, x_C^{k_1}, x_C^{k_2}, ..., x_C^{k_i}\}$
4. Update the best solution in the reject-solution set $x_{k_i}^* = \arg\min_{x\in\Psi_{k_i}} \overline{G^{N_{k_i}}(x)}$
5. Acceptance/Rejection
   Compute $\rho^{k_i} = \frac{\overline{G^{N_{k_i}}(x^{k_0})} - \overline{G^{N_{k_i}}(x_{k_i}^*)}}{r^{k_i}(0) - r^{k_i}(d_{k_i}^*)}$
6. Trust Region Update
   if $\rho^{k_i} < \eta_0$, $\triangle_T^{k_{i+1}} = \gamma_1 \triangle_T^{k_i}$, $\triangle_S^{k_{i+1}} = \gamma_1 \triangle_S^{k_i}$, let $\triangle_T^k = \triangle_T^{k_{i+1}}$, $\triangle_S^k = \triangle_S^{k_{i+1}}$, increment i by 1 and go back to step 1.
   elseif $\eta_0 < \rho^{k_i} < \eta_1$, $\triangle_T^{k_{i+1}} = \triangle_T^{k_i}$, $\triangle_S^{k_{i+1}} = \triangle_S^{k_i}$, let $\triangle_T^k = \triangle_T^{k_{i+1}}$, $\triangle_S^k = \triangle_S^{k_{i+1}}$ increment i by 1 and go to step 7.
   else $\rho^{k_i} > \eta_1$, $\triangle_T^{k_{i+1}} = \gamma_2 \triangle_T^{k_i}$, $\triangle_S^{k_{i+1}} = \gamma_2 \triangle_S^{k_i}$, let $\triangle_T^k = \triangle_T^{k_{i+1}}$, $\triangle_S^k = \triangle_S^{k_{i+1}}$ increment i by 1 and go to step 7.
7. "Sufficient reduction" test: Do the sufficient reduction hypothesis test with type I error α_k. If the test is passed, increment k by i, let $x^{k_i} = x^{k+1}$ and go back to the main algorithm step 1 of Stage II. Otherwise, increment i by 1 and go to step1.

Figure 2 illustrates the framework of sub-algorithm of STRONG. In the sub-algorithm, only quadratic models are used to approximate the underlying function. The sub-algorithm will work on enhancing the fidelity of the local model and reducing the sampling error by increasing the replication for the center point as well as adding more design points within the sampling region sequentially. The computational efforts will be adaptively increased until a satisfactory solution is obtained. Once the satisfactory solution is obtained, STRONG will go back to the main algorithm and move to the improved solution. The number of replications for center point as well as the number of design points within the sampling region will go back to the originally fixed level which is specified in the main algorithm.

There are three main reasons that may cause the rejection of the solution in stage I (i.e., produce the unsatisfactory solution).

1. The local approximation model is poor.
2. The step size is too large so that the predicted reduction does not match the true reduction.
3. Sampling error of the response function.

The sub-algorithm tackles these situations by taking the corresponding solid strategies.

1. Shrink the sampling region, increase the number of replication for the center point and add more design points.
2. Shrink the trust region.
3. Collect all visited solutions within the trust region and increase the replication for each of them.

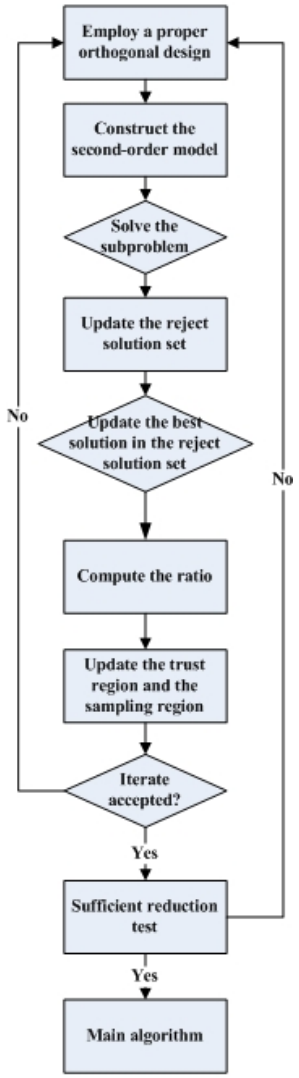

Figure 2: Framework of the sub-algorithm.

In the first strategy, with the shrinkage of sampling region, the number of replications for the center point is increased. The response sampling error for the center point will decrease. Thus, the intercept estimate will be more and more precise. Meanwhile, the design points are also accumulated. The gradient estimate can be shown to be more and more precise as well in this fashion. In other words, the intercept and the gradient estimate in the quadratic model will be gradually approaching the true value. In the second strategy, the optimization step will be more restrictive with the shrinkage of trust region. STRONG will rather be more conservative in optimization step if the current local model cannot provide a satisfactory solution, i.e., the local model is not reliable. In the last strategy, the visited points will continue updating when more replications are taken for each of them since the visited solutions could be wrongly rejected due to randomness in the previous iterations. This strategy allows the visited and wrongly rejected solutions to come back if it finally turns out to be a good solution.

Sub-algorithm will keep shrinking the trust region as well as the sampling region by the same ratio until the solution is accepted. The model fidelity can be improved by the arguments we made earlier. On the other hand, the bias between the metamodel and true underlying response function will be reduced. In both main algorithm and sub-algorithm, STRONG uses "trust region" to control the step size. It is an adaptive and systematic way to manage the optimization step. STRONG won't be too conservative to proceed when the model approximation does a good job of predicting improvement. Moreover, it is automatically determined in the algorithm. In STRONG, the trust region and the sampling region are deliberately separated from each other. The reason is because the trust region controls the optimization step while the sampling region is related with the gradient estimate. Therefore, the trust region is expected to be large so that the optimization step won't be restricted in a small region, whereas the sampling region is required to be small so that a better gradient estimate can be obtained. Following the framework of sub-algorithm, we made the following observations.

1. For each iteration (no matter in main algorithm or in sub-algorithm), the algorithm implied that the sampling region is contained in the trust region, i.e., $\mathscr{S}^k \subset \mathscr{T}^k$
2. The sample size for each one in the reject-solution set is increasing with iterations and $N_{k_1} < N_{k_2} < \ldots < N_{k_i}$
3. $\Psi^{k_0} \subset \Psi^{k_1} \subset \ldots \subset \Psi^{k_i}$
4. $\lim_{i\to\infty} G^{N_{k_i}}(x^{k_o}) = g(x^{k_0})$
5. The quadratic approximation model γ^{k_i} changes with the iteration k_i (i.e., $\widehat{g}(x_C^{k_i})$ and $\widehat{\nabla} g(x_C^{k_i})$ change with iteration).

3.5 Theoretical Performance

We have proved the following results.

Theorem 1. *$\forall$ iteration k, if the local approximation model is given by $\gamma^k(d_C^k) = \widehat{g}(x_C^k) + \widehat{\nabla} g(x_C^k) d_C^k + \frac{1}{2} d_C^{kT} \widehat{H}(x_C^k) d_C^k$ then the Cauchy step d_C^k satisfies*

$$\gamma^k(0) - \gamma^k(d_C^k) \geq \frac{1}{2} \parallel \widehat{\nabla} g(x_C^k) \parallel min(\frac{\parallel \widehat{\nabla} g(x_C^k) \parallel}{\parallel \widehat{H}(x_C^k) \parallel}, \triangle_T^k)$$

Theorem 2. *If the sub-algorithm generates a subsequence of iterate points* $\Psi^{k_i} = \{x_C^{k_0}, x_C^{k_1}, ...x_C^{k_i}\}$ *then* $\exists$ *trust region* $\triangle_T^K$ *and some constant* C_2*, s.t., when* $\triangle_T^{k_i} < \triangle_T^K$, $\| \gamma^{k_i}(d_C^{k_i}) - \overline{G^{N_{k_i}}(x_C^{k_0} + d_C^{k_i})} \| \leq C_2(\triangle_T^{k_i})^2$

Theorem 3. $\forall x^{k_0}$ *in the sub-algorithm, if* $\| \nabla g(x^{k_0}) \| \geq \xi > 0$*, then* $\exists K$*, s.t. whenever* $k_i \geq K$*,* $\triangle_T^{k_i} \leq \triangle_T^K$*,* $\exists d^{k_i}$ *we eventually have* $\rho^{k_i} \geq \eta_0$.

Theorem 4. *For any initial point* x^0*, the algorithm will generate a sequence of iterate solutions* $\{x^k\}$ *in the main algorithm, assuming the last subsequence of iteration solutions in the sub-algorithm are* $\{x^{k_i}\}$ *then either* $\lim_{k\to\infty} \| \nabla g(x_C^k) \| = 0$ *or* $\lim_{i\to\infty} \| \nabla g(x_C^{k_i}) \| = 0$

Theorem 1 provides an lower bound of the reduction of the local model for each iteration. Notice that this lower bound depends on the local model. Theorem 2 guarantees that in sub-algorithm if the trust region keeps shrinking then eventually the difference between predicted response and the average of predicted response will be bounded by the square of trust region size. Theorem 3 provides the guarantee that the sub-algorithm will never stop unless we are at a stationary point. In particular, it guarantees that a good step d^{k_i} will be eventually accepted if the current iterate is not stationary (i.e., $\| \nabla g(x^{k_0}) \| \geq \xi > 0$), and a new iterate $x^{k_{i+1}}$ will be generated. Theorem 4 provides the convergence guarantee of STRONG.

4 PRELIMINARY NUMERICAL RESULTS

Consider the well-known Rosenbrock test function in the optimization literature. This test function was first presented in Rosenbrock(1960). This test function has an interesting shape that the solution lies in a curved valley (Spall 2003).

$$g(x) = \sum_{i=1}^{p/2} [100 * (x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2] \qquad (7)$$

Without losing the generality, we assume $p = 2$ and use Monte-Carlo simulation to evaluate the response value

$$G(x) = 100 * (x_2 - x_1^2)^2 + (1 - x_1)^2 + \varepsilon, \qquad (8)$$

where $\varepsilon \sim^{i.i.d.} N(0, \sigma^2)$.

Notice that the minimum solution is $(x_1, x_2) = (1, 1)$ and the minimal objective function value is 0. The following numerical evaluation will focus on the comparisons between STRONG and FDSA (Finite-Difference Stochastic Approximation). It is notable that $g(x)$ is the objective function and x^* is the true optimal solution $(1, 1)$ and $g(x^*) = 0$, $g(x_C^k) - g(x^*)$ represents the distance of the current solution to the optimal solution. In the following cases, we will use 2^k factorial design to fit the linear model and use central composite design to fit the quadratic model.

Case 1: The starting point is set at (30, -30) and the variance of noise is set at 10. STRONG (n=2, N=5) and FDSA are used to solve the problem.

Table 1: Performance of STRONG in Case 1

No. of obs.	$g(x_C^k) - g(x^*)$	Distance Reduced
0	86490841	0%
120	1680400	98%
300	49351	100%
420	11921	100%
530	62.84	100%
600	24.24	100%
680	1.58	100%

Table 2: Performance of FDSA with the bad starting point in Case 1

No. of obs.	$g(x_C^k) - g(x^*)$
0	86490841
10	9801000000
20	9801000000
100	9801000000

Case 2: The starting point is set at $(3, -3)$ and σ^2 is set at 10, bound of parameter is $(0, 5) * (0, 5)$.

Table 3: Performance of FDSA with the good starting point in Case 2

No. of obs.	$g(x_C^k) - g(x^*)$	Distance Reduced
0	3604	0%
20	1566.7	98%
40	52.02	100%
60	1.01	100%
80	0.82	100%
100	0.77	100%

Case 3: Follow the same setting in Case 1, we compare the numerical performance of STRONG in different variance. The stopping rule is set at $g(x_C^k) - g(x^*) < 1$.

Table 4: Performance of STRONG with different variance in Case 3

Variance	No. of obs.
1	2732
10	3256
20	6242
30	12812
100	18148

We compared the numerical performance between STRONG and FDSA. Case 1 shows that STRONG can converge to the optimal solution in less than 700 function evaluations even when staring point is far from the optimal solution (see Table 1). In addition, in the first 120 observations, the distance from starting point to the optimal solution

is reduced by 98% which illustrates its fast convergence rate. Case 2 also shows that FDSA can be divergent (see Table 2) if the same far starting point is applied in FDSA. This case shows that FDSA is very sensitive to the starting point. Case 2 illustrates that FDSA could converge and is more efficient if the starting point is chosen close to the optimal solution. Notice that in Case 1 (FDSA) we do not restrict the parameter space but in Case 2 parameter space for FDSA is restricted in a small region (0,5)*(0,5). Case 3 shows STRONG can still converge to the optimal solution even when the variance of noise is set very high (see Table 4), whereas FDSA will still diverge since the setting is the same as Case 1.

5 CONCLUDING REMARKS

This paper discussed a novel RSM-based simulation optimization method. Compared to traditional response surface method, STRONG is automated and has been proved to converge to the local optimal solution with probability 1. Besides, STRONG has a flexible and "smart" sequential strategy for constructing the metamodels which can mitigate the computational loadings greatly. In the numerical evaluation, we show that STRONG outperforms FDSA algorithm which may diverge when the response is very steep (Andradottir 1995). STRONG, on the other hand, can efficiently converge to the optimal solution regardless of the location of the starting solution. Furthermore, STRONG is pretty flexible in selection of experimental designs, combined with a efficient factor screening method, STRONG has the potential to solve high-dimensional problems, which we will explore in the future research.

ACKNOWLEDGMENT

This research was partially supported by Purdue Research Foundation (PRF), Naval Postgraduate School Award N00244-06-C-0002, and Hong Kong Research Grants Council grant CERG 613305.

REFERENCES

Andradottir, S. 1995, Nov.. A stochastic approximation algorithm with varying bounds. *Operations Research* 43 (6): 1037–1048.

Barton, R., and M. Meckesheimer. 2006. *Handbooks in operations research and management science: Simulation*, Chapter 18, 535–574. Oxford: Elsevier.

Box, G., W. Hunter, and J. Hunter. 1978. *Statistics for experimenters: An introduction to design, data analysis and model building*. New York: Wiley.

Box, G., and K. Wilson. 1951. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society* B13:1–45.

Carson, Y., and A. Maria. 1997. Simulation optimization: Methods and applications. *Proceedings of the 1997 Winter Simulation Conference*:118–126.

Conn, A., N. Gould, and P. Toint. 2000. *Trust-region methods*. SIAM.

Fu, M. 1994. Optimization via simulation:a review. *Annals of Operations Research* 53:199–247.

Fu, M. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14:192–227.

Khuri, A., and J. Cornell. 1996. *Response surfaces: designs and analyses*. New York: Marcel Dekker.

Myers, R., and D. Montgomery. 2002. *Response surface methodology*. New York: Wiley.

Nicolai, R., R. Dekker, N. Piersma, and G. Oortmarssen. 2004. Automated response surface methodology for stochastic optimization models with unknown variance. *Mathematics and Operations Research*:491–499.

Nocedal, J., and S. Wright. 1999. *Numerical optimization*. Springer.

Robinson, S. 1996. Analysis of sample-path optimization. *Mathematics of Operations Research* 21:513–528.

Spall, J. 2003. *Introduction to stochastic search and optimization*. New York: Wiley.

AUTHOR BIOGRAPHIES

KUO-HAO CHANG is a PhD candidate in the School of Industrial Engineering at Purdue University. He received a B.B.A. in Statistics from National Cheng Kung University and a M.S. in Industrial Engineering and Engineering Management in National Tsing Hua University. His research interests include simulation optimization, stochastic modeling and financial engineering. His email address is <changk@purdue.edu>.

L. JEFF HONG is an assistant professor in industrial engineering and logistics management at The Hong Kong University of Science and Technology. His research interests include Monte-Carlo method, sensitivity analysis and simulation optimization. He is currently associate editors of *Naval Research Logistics* and *ACM Transactions on Modeling and Computer Simulation*. His e-mail address is <hongl@ust.hk>.

HONG WAN is an Assistant 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>.